Performability Analysis and Validation of a Large Scale Fieldbus System by Formal Methods

Von der Fakultät für Maschinenbau

der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde

eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von

M.Sc. Jieyu Liu
aus Taiyuan, Shanxi, China

eingereicht am: 06.06.2016
mündliche Prüfung am: 16.09.2016

Gutachter:
1. Referent: Prof. Dr.-Ing. Dr. h.c. mult. Eckehard Schnieder
2. Referent: Univ.-Prof. Dr.-Ing. Alexander Fay

2016

Acknowledgments

This dissertation is based on my research work in das Institut für Verkehrssicherheit und Automatisierungstechnik (iVA), Technische Universität Braunschweig, Germany for the last six years. It has provided me comprehensive descriptive means, methods and tools for analyzing complexity and performability of a system and its components. Without kind support from many people, this work would not have been completed.

Above all, I would like to express sincere gratitude to my doctoral supervisor, the former director of das Institut für Verkehrssicherheit und Automatisierungstechnik, Technische Universität Braunschweig, Prof. Dr.-Ing. Dr. h.c. mult. Eckehard Schnieder, who has not only accepted and allowed me to pursue my doctoral research stay at iVA, but also enlightened and encouraged me of choosing the direction and proceeding my research during the entire doctoral period.

I am grateful to Prof. Dr.-Ing. Alexander Fay, the director of Institut für Automatisierungstechnik, Helmut-Schmidt-Universität as my second dissertation assessor. I appreciate Prof. Dr.-Ing. Uwe Becker, the current director of iVA, as the chairman for hosting my doctoral defense.

Moreover, I would like to thank Dr.-Ing. Harald Schrom for designing and developing the SmallCAN system, the dynamic behavior of which my modeling and validation work are combined with. I also want to mention Patrick Diekhake for his team-work. I would not hesitate to thank all my colleagues at iVA and all my friends, whoever supported and accompanied me through my research period.

Furthermore, I am indebted to CSC (Chinese Scholarship Council), providing the financial support during the first four years of my research stay. I am also grateful to iVA for the adequate support of the rest years of my research stay.

Finally, I would like to express my deepest thankfulness to my family because of their comprehensive tolerance and overwhelming support: my mother Hao, Taili; my father Liu, Huimin; my mother-in-law Sun, Fengxian; my father-in-law Fu, Chunshan; my dear wife Fu, Tingting and the last but not least, my beloved sons: Liu, Boqian and the new comer, Liu, Bosong.

Abstract

The complexity of large scale fieldbus systems is two-fold: message-sending concurrency and emergent bus behavior. On the one hand, an increase in the number of accumulating nodes within one fieldbus system expands its message-sending concurrency; on the other hand, the growth of emergent bus behavior causes a temporary or lasting message burst on the fieldbus channel. The message sequences in turn have an increased burst behavior, aggravating the traffic density. Therefore, this dissertation evaluates the performability of large scale fieldbus systems by presenting a busload validation procedure by formal methods.

The model concept is conceptualized and formulated by UMLCD and OSI Model. Furthermore, the validation procedure is formalized and structurally specified by applying the attribute hierarchy and BMW principle. Based on sorting the message-sending occurrences from the log data of a real fieldbus-based building automation system, the validation procedure is thus quantified with the real system timed-parameters. In addition, the stochastic distributions of message transmissions are determined by the goodness of fit method.

The entire work is based on DSPN as formal means of descriptions and models. The corresponding Petri net communication model is hierarchically constructed, which has been further parameterized, integrated and simulated.

The analysis of system complexity is provided by the programming-based extension of the Petri net communication model. In addition, the results of Monte-Carlo-Simulation have been sorted, analyzed and evaluated regarding the validation aspects of system performability. Finally, the emergent message burst generated from the function interrelations has also been observed and evaluated. The result of this work will make a formal contribution to the improvement the fieldbus specification.

Kurzfassung

Insbesondere für Feldbussysteme mit einer großen Anzahl an Busteilnehmern wird die Komplexität über zwei Kenngrößen charakterisiert. Einerseits stellt die Erhöhung der Anzahl akkumulierter Feldbusknoten innerhalb eines Feldbussystems eine gestiegene Message-Sendung-Nebenläufigkeit dar. Andererseits steigt diese auch durch Zuwachs des emergenten Busverhaltens, die temporäre oder dauerhafte Nachrichtenfolgen mit sich führen. Die Nachrichtenfolgen wiederum können ein erhöhtes Burst-Verhalten auf dem Feldbuskanal, d.h. eine erhöhte Busauslastung verursachen. Ziel der vorliegenden Arbeit ist es, ein komplexes Feldbussystem formal zu beschreiben und ein formales Buslastvalidierungsverfahren darzustellen.

Das Modellkonzept wird zunächst durch das UMLCD und das OSI-Modell formuliert, und anschließend wird das Validierungsverfahren mit der Attributhierarchie und dem BMW-Prinzip formalisiert und spezifiziert. Aufgrund der Sortierung des Sendungsverhaltens mittels Logdaten eines realen Feldbus-basierten Gebäudeautomationssystems, wird das Validierungsverfahren durch die quantitative Analyse weitergeführt. Zusätzlich werden die stochastischen Verteilungen der Sendungsverhaltene durch die Goodness-of-Fit Methode angepasst.

Die gesamte Arbeit basiert auf DSPN als formales Beschreibungsmittel und Modellierungsmittel. Das entsprechende Petrinetz-Kommunikationsmodell wird vorgestellt, welches hierarchisch konstruiert, parametriert und simuliert wurde.

Die Systemkomplexität wird mit Hilfe der Programmierung-basierten Erweiterung des Petrinetz-Kommunikationsmodells analysiert. Dazu werden die Monte-Carlo-Simulationsergebnisse dieses erweiterten Modells vorgestellt, analysiert und bewertet und in Bezug zu den Validierungsaspekten der Systemleistung gesetzt. Schließlich wird das erzeugte Nachrichten-Burst-Verhalten von den Funktionsverknüpfungen beobachtet und bewertet. Die Ergebnisse werden von dieser Arbeit nach der Vervollständigung der formalen Feldbusspezifikation zurückgeführt und verbessert.

Contents

Chapter 1

Introduction

1.1 Background

According to IEC (International Electrotechnical Commission) 61158, a Fieldbus is defined as an open, digital and multi-drop communications network system with intelligent field devices for real-time distributed control [IEC61158:3-6, 2000]. It provides an single and efficient connectivity among all fieldbus participants. Fieldbus systems have been being standardized with various specifications for decades due to their comprehensive adoptions in industrial and academic applications.

The advantages of fieldbus applications are obvious: savings in wiring as well as the enhanced multi-computer system performance. In addition, the low cost, high energy efficiency, savings in operating costs and reduced downtime are also ascribed to fieldbus networking topology. The failure isolation among nodes also increases system dependability and simplifies the configuration and maintenance. Compared with wireless networking solution, fieldbus systems have no interference problems and are less dependent on power supply [Güngör et al., 2011].

Standardization, research and applications in fieldbus have been widely carried out more than 30 years. [Kopetz, 2011] introduces requirements and validations for distributed embedded applications . [Sauter et al., 2000] give an overview of fieldbus systems, including its history, properties and developing tendencies.

For theoretical investigation of fieldbus systems, different research methods have been applied. One is based on modeling with Petri nets, which have been applied on several computer and communication systems.

Most early works focus on evaluating the QoS (Quality of Service) for communication systems, such as real-time and robustness. Meanwhile applying Petri nets in performance evaluation of computer systems has been widely carried out [Ajmone Marsan et al., 1984]. The works of [Lindemann, 1998], [Molloy, 1982], and [Ramamoorthy and Ho, 1980] have discussed the model-based performance analysis by means of DSPN (Deterministic and Stochastic Petri Nets). A model-based design and methodology of validating and analyzing the performance of automotive control system is provided by applying their own simulation tools [Hanzlik and Kristen, 2013]. [Chen et al., 2012] has introduced a verification approach of communication protocol by using CPN (Colored Petri net).

1.2 Motivation

The tendency of fieldbus technology is to consistently get "more intelligent" with more interactive communications, providing larger scale number of fieldbus nodes [Kiefer, 1996]. Due to information explosions as well as an increase in demand for system performability, the complexity has been rising especially in fieldbus systems with large scale numbers of bus nodes.

In this work, the property complexity, threatening the performance and availability of fieldbus systems, can be categorized into two characteristics: the message-sending concurrency and emergent bus behavior. The message-sending concurrency together with its concurrency resolutions integrated in DLL (Data Link Layer) is discussed under the maximum spatial networking conditions in one fieldbus system. An increasing number of accumulating nodes within one fieldbus system accumulates the sending concurrency. By contrast, the functional complexity is mapped with the function relations defined in the APL (Aapplication Layer) of the OSI model (Open System Interconnection model), generating a temporary or lasting message sequence onto the bus channel.

It continuously burdens the bus channel until these function solutions have been completely executed. The motivation of this validation approach is hereby presented based on quantitative analysis of system complexity.

In addition, this tendency of more intelligence in fieldbus struggles with its Achilles' heel: relatively low bit rate (compared with paralleled communication topology), limited wiring length and a increasing number of networking nodes and their function interactions. This new tendency of fieldbus systems might have an impact on system performance or even lead to unavailability. Therefore, a thorough understanding of a complex fieldbus system's behavior plays an important role of fulfilling the performance requirements.

This dissertation is based on a platform denoted as SmallCAN. It is a fieldbus-based application in building automation with low power and low cost [Schrom, 2003], which is designed and developed by iVA *das Institut für Verkehrssicherheit und Automatisierungstechnik*, Technische Universität Braunschweig, Germany.

According to the V-model of system development, SmallCAN is currently nearing the end of system implementation phase and has also entered the system validation phase. As a fieldbus-based system in building automation, bus nodes in charge of functions, such as HVAC (Heating, Ventilation, and Air Conditioning), safety and security control inside every building unit, are spatially integrated into one fieldbus. As a result, it is inevitable that SmallCAN incorporates a considerable number of bus nodes as well as the numerous functional interactions among them. As a result, it fits the concept of a complex fieldbus system. Therefore, performability analysis and validation need to be proceeded.

In order to profile and evaluate the complex behavior of SmallCAN, a computerized modeling is necessary. A model is a simplification or an abstraction of reality [Booch et al., 1999]. The model is built with those characteristics and quantities impacting on the performability of the whole system. In addition, the accuracy between the model and the complex system is improved by integrating quantitative results acquired from the real system into the model.

One of the formal means of modeling is Petri net. Its mathematical and graphical capabilities are well suited for profiling the concurrency in communication systems. The stochastic timed characteristic of the system complexity can be modeled and analyzed in DSPN [Liu et al., 2013]. It is hereby applied in this work for modeling and mapping SmallCAN as a complex fieldbus system with large number of concurrencies and message bursts.

The model may also be analyzed with simulation. A computerized model emulation can generate simulation results, evaluating the system complexity. Therefore, the motivation in this work is to present a validation procedure by formal methods regarding performability of SmallCAN as a complex fieldbus system.

Based on SmallCAN related projects, the system has already been implemented in several real buildings scenarios under diverse costumers' requirements. Two main problems concerning analyzing the system complexity emerged as follows:

1. The upper boundaries, such as maximum cable length up to 1000 meters and maximum number of bus nodes up to 1000, quantify SmallCAN's message-sending concurrency.
2. For the potential development of SmallCAN, possible functions are interconnected with diverse combinations. Emergent bus behavior can be interpreted as these possible function connections and interactions. As a result, temporary or even lasting message sequences are generated. This might consequently prolong the entire execution time of those function-related messages. Moreover, the consequences caused by functional complexity could be severe, such as message lost

and function defect. [Schrom et al., 2011] have defined and implemented these functional interrelations in a real fieldbus-based building automation system.

Therefore, analyzing the system complexity for perfomrance evaluation of Small-CAN as case study of large scale fieldbus systems by formal methods plays an important role in protocol improvement and further system standardization.

According to the railway norms of communication safety [EN50129, 2003], the types of message errors in closed transmission systems can be correspondingly interpreted as: long-time burst (continuous concurrency) and short-time burst (message sequence).

The necessity of research in fieldbus system performance depends on the timing specifications as well as the control-oriented targets. A good case study is the comparison between vehicle fieldbus system and fieldbus-based building automation system. The delays of important function-executing-messages, such as emergency stop and speed control, are intuitively intolerant in vehicle fieldbus requirements. By contrast, the delays occurred in fieldbus-based building automation system seem acceptable. Such as the delays of messages switching lamps are intuitively allowed. It seems that the reliability of fieldbus-based building automation systems has not been impacted, even if the lights are switched seconds-wise later [Thomesse, 2005]. However, this conclusion is based on the limited complexity existed in fieldbus systems.

The expansion of system complexity has been universally inevitable, which leads to impacts on the system performability. If the function interrelations among messages are unlimited defined, the message sequencing increasingly aggravates the busload, which causes burst error. So the burst error is described as the type of message sequence error in which the messages executing the similar functions are arranged to transmit on the bus channel. This might lead to message disorder or message deletion. Message disorder is defined that the order of the message sequence during message stream is changed, whereas the message deletion represents that the message is unintentionally removed from the message stream. In addition, retransmissions usually occurs in the bus nodes assigned with low sending priorities, which also burdens the bus channel under the condition of higher traffic density. Finally, any scenario mentioned above could lead to function disability and hazard situations under high degree of system complexity.

In order to investigate the impacts of complexity on fieldbus systems, different busload types together with their different access mechanisms need to be investigated, although the comparison between the communication performability and requirement's data of these process are very labor-consuming [Kiefer, 1996].

On one hand, the high complexity degree in fieldbus systems can be characterized by the non-deterministic behavior of concurrent message transmissions. Besides, the complexity also attributes to strong dependencies between fieldbus components and their environment [Kaneshiro et al., 2007]. A good case study shows that in fieldbus-based building automation systems, the applications are more relevant to data acquisition and supervision than their function control [Thomesse, 2005].

On the other hand, more function interrelations are defined inside the the Application Layer of the OSI model. They can be mapped along with more emergent messages generated as message burst. This occurs from the aspect of bus channel in closed transmission systems. It continuously burdens the fieldbus channel and accumulates the busload until these functions are fully executed.

Therefore, a performability analysis, profiling and evaluating the structural and functional complexities of such large scale fieldbus systems, is necessary for providing a validated criteria for further system development of SmallCAN.

1.3 Previous iVA-related Works

The work of [Schnieder, 2010] has presented a comprehensive discussion of model concepts. Focusing on a complex fieldbus system and its specified the fieldbus requirements, the model concept is formulated and formalized in this work.

[Kiefer, 1996] has provided a careful review of fieldbus types and their requirements regarding process control. Based on this, a performance analysis approach has hereby carried out by colored Petri nets. By contrast, DSPN is applied in this work as one of the modeling and simulation means.

SmallCAN, as a fieldbus-based building automation system, has been developed and thoroughly discussed in [Schrom, 2003]. Based on this, the emergent busload behavior in this work can be further analyzed and validated with formal methods.

In this work, the mechanisms resolving the message concurrency, such as CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) and bitwise priority comparison, are quantified and later integrated into the model-based validation procedure.

[Diekhake, 2016] adopts causal Petri net and later bind the Petri net model with the real fieldbus system, providing a platform of on-line supervision and simulation. As a result, the focus of the predefined causality of the functional structure described with incidence matrix can be tested in real-time. In addition, the response time of transmitting a message from a source and a drain is also measured by this way.

In this work, the focus of increasing fieldbus complexity is categorized and then analyzed repetitively, based on which a timed busload validation procedure is proceeded. Each step of this validation procedure with its timed requirement is formally specified and assigned in detail, including the scenarios of channel occupying, medium accessing and concurrency handling. In addition, A DSPN model is constructed and then parameterized with the fitted stochastic distribution types of message-sending occurrences, sorted from the log data in real fieldbus system.

Moreover, the validation procedure in this work includes not only the characteristics and quantities of mapping between Data Link Layer and Application Layer of OSI model, but also the interactions between the fieldbus system and its environment. Finally, the DSPN-based busload validation procedure is completed by the transition firing rates-based evaluation and observation with the aspect of system performability.

1.4 Structure and Description of the Dissertation

This dissertation consists of 8 Chapters, providing a formal validation procedure of a large scale fieldbus system. First by conceptualizing, formulating and formalizing the structural and emergent bus behavior of the large scale fieldbus system. Second by classifying and quantifying the concurrent message-sending; third by modeling and parameterizing the concurrent message-sending scenario. Followed by scaling up from establishing a simple communication model to extending model with adjustable traffic densities. Then by analyzing and evaluating transition firing rates based on Monte-Carlo simulation results of Petri net models, completing a busload validation procedure, as is shown by Figure 1.1.

Chapter 1 introduces the background of the focused research field, the motivation, the previous works and the structure of this dissertation.

Chapter 2 presents the general requirements of different fieldbus applications related in this work. Based on this, fundamental backgrounds of descriptive means in this dissertation are hereby presented, such as UMLCD (Unified Modeling Language Class Diagram), OSI model, Petri net theories, including the formal definitions of SPN (Stochastic Petri Net) and applied probability theory. Third, methods of modeling concepts and system hierarchical abstraction are selected for conceptualizing the concept model. Then, methods of system context, system properties and attribute hierarchy are hereby introduced for formalizing the validation procedure of the concept model. Besides, method of probability distribution fitting is also presented for quantifying the message-sending behavior. In addition, scenario arts, such as worst case scenario, and computerized methods, such as adjustable model extension and steady state analysis for model simulation, are introduced respectively.

Chapter 3 formulates and formalizes the SmallCAN specifications by applying attribute hierarchy method to structuring the subsystems in OSI model, such as APL, DLL and the bus channel. Then, the synchronization of these subsystems provides an overview of assignments concerning busload validation. Moreover, these focused system properties are further the categorized as the elementary properties and the emergent properties. The emergent property generated from the interlinking between a communication system and its means of formal descriptions are specified by the mapping between UMLCD and DSPN. Finally, the validation procedure focusing on the system complexity is structured and specified in detail.

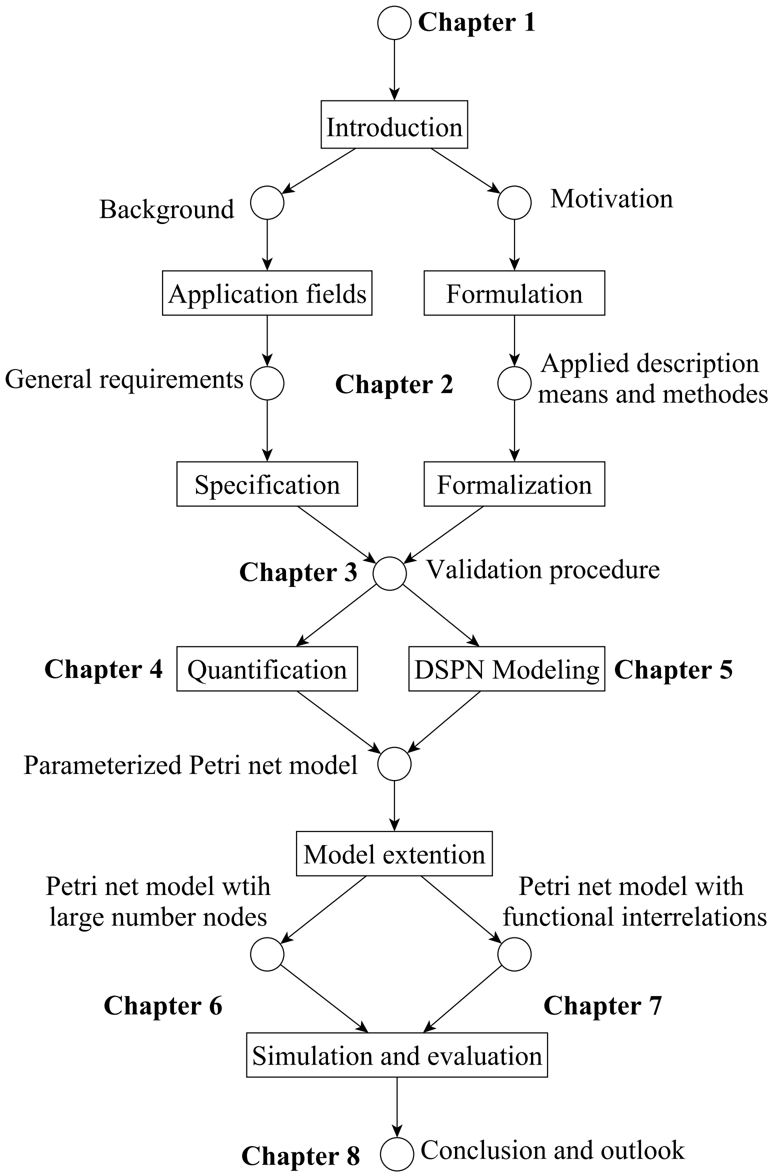


Figure 1.1: **The structure and processes of the dissertation in Petri net description**

Chapter 4 quantifies the requirements related to SmallCAN busload generation and resolution. First, the dynamic behavior between the SmallCAN system and its context has been characterized by the attribute hierarchy. Based on this, a quantitative analysis of the selected mechanisms in MAC, such as CSMA/CA and bitwise arbitrary, has been carried out. Second, for the quantitative analysis in real SmallCAN system, the temperature related message-sending behavior has also been profiled by plotting the relevant SmallCAN log data. Moreover, time interval of two typical message IDs with their relevant PDFs (Probability Density Functions) have been further abstracted and sorted. Moreover, by fitting the stochastic behavior of message-sending events, the goodness of fit method has been applied to fitting distribution types of the stochastic behavior of the message-sending in real SmallCAN system given by the most likelihood values, including fitting the quantities, such as expectations and variances. Based on this, channel concurrency has been theoretically defined and mathematically deducted based on the fitting results.

Chapter 5 focuses on the complexity generated from the large scale fieldbus system. providing a model-based implementation approach for validation. First, the net structure and modeling assignments have been outlined. Second, the DSPN modeling environment together with its analytical features, in this work the software named π-Tool, is hereby introduced. Third, the modeling-based implementations and integrations of the concurrent message-sending scenarios are hierarchically implemented into two kinds of sub-models, the channel state sub-model and the message-source-node sub-model. In addition, the interconnections together with the selected DSPN modeling elements are listed and explained in detail. Fourth, the concurrency handling mechanism, such as CSMA/CA and bitwise arbitrary mentioned in Section 4.4, have been integrated into the hierarchical Petri net communication model. According to the probability distribution fitting method mentioned in Section 2.3.4, the timed parameters of the message-sending frequency extracted and sorted from the log file of the fieldbus-based building automation system, shown in Section 4.6, are hereby parameterized with the transition firing expectations as model input. This is done in each message-source-node sub-model, as is shown in Section 5.4. Fifth, the quantified parameters are hereby set in the transition firing expectations as model input. Finally, the model has been checked with token flow function, performing in a satisfactory manner under low traffic conditions.

Chapter 6 has carried out the performance and real-time availability evaluation under the outlined SmallCAN performability assignments. The transition firing rates are generated from the Monte-Carlo simulation. First, a real-time availability approach is introduced regarding analyzing the concurrent-message-sending. Second, the performance of parameterized DSPN model has been evaluated under the conditions of low and high traffic density. The channel collision scenario has been hereby validated with the aspect of dependability. The quantities influencing the performability and the selected access mechanisms, such as CSMA/CA and bitwise priority comparison, have been analyzed and evaluated respectively. The extended communication model with these incorporated quantities has been analyzed, not only to validate the correctness of the communication model structure, but also to provide formally validated criteria for further fieldbus protocol development.

Chapter 7 focuses on the emergent message burst caused by functional relations. First, function interrelation regarding the emergent property Definition has been formally defined. Second, a case study focusing on the emergent message burst has been proposed, functional relations of which are abstracted from a real SmallCAN system in building automation. Third, the quantities of generated function-based message sequence has been measured and discussed, such as time intervals and typical function paths predefined inside the APL. Furthermore, these function paths in the case study have been hierarchically structured using the defined term function-interrelation-depth. With different function-interrelation-depths, the interactions and function relations among involved message-source-nodes are discussed by UMLCD and later integrated to the extended Petri net communication model.

The token animation function inside π-Tool is hereby applied to generate the dynamic behavior of the Petri net model. By observing the focused transition firing sequences of two function interrelation depths involved, the time intervals of the these focused token flow have been observed and profiled.

With the aspect of emergent message burst, this observation approach can be applied as an upper criterion of defining the maximum amount of functional relations as well as function interrelation depths inside the Application Layer. As a result, emergent message bursts are restrained to occupy on the bus channel.

An analysis with a case study of adjustable structuring the function interrelations by selecting different temperature message sequence is presented and described in detail. By integrating the extended communication model with variable function-interrelation-depths, the analysis of observed transition firing rates has also been discussed in detail.

Chapter 8 concludes this validation procedure and covers the future work.

Chapter 2

Adapted Description Means and Methods regarding Complexity

In this chapter, the prerequisites for evaluating and validating complex fieldbus systems are listed. Based on this, suitable description means and methods are selected and introduced.

The formal means of description are constructed with a precise set of symbols and tolerable sequence order of these symbols [Schnieder et al., 2009]. Selected description means in this dissertation are hereby gradually introduced. First, UMLCD and selected relationships are presented for structuring a focused concept and its entirety in Section 2.2.1. Second, OSI model are applied in describing the focused subsystems and their properties in a large scale fieldbus system named SmallCAN in Section 2.2.2. Third, the detailed discussion of the motivation and definitions of applied Petri net are presented in Section 2.2.3. Then, applied definitions concerning probability theory are introduced in Section 2.2.4.

Applied methods in this dissertation are then discussed as follows. First, system abstraction hierarchy is applied in formulating the model concepts in Section 2.3.1. Second, system properties and system context are introduced in Section 2.3.2. Third, attribute hierarchy is presented for formalizing the system model in a derivative manner in Section 2.3.3. Then, computerized method is applied and introduced for steady state analysis in Section 2.3.6. Moreover, probability distribution fitting method together with its motivations is discussed in the context of analyzing the interactiveness between the complex fieldbus system and its environment in Section 2.3.4. Finally, Scenario arts are introduced for busload modeling in Section 2.3.5.

UMLCD is applied as means of description for structuring a focused concept and its entirety. Petri net is introduced as means of realization for communication modeling. Applied definitions and methods related to probability theory are hereby presented as prerequisites for further quantitative analysis, such as applying the goodness of fit method for fitting the stochastic distribution type.

2.1 Requirements of Different Fieldbus Types

As for specific application fields, the first-generation fieldbus was put into practice and eventually standardized with various application purposes. CAN protocol was founded by the Robert Bosch GmbH in Germany originally for automobile [Bosch, 1991]. CAN's bit rate is inversely proportional to its networking length. It is used not only for the power-train control, but also the reliable and ambient system inside and outside the vehicle. Another widely used fieldbus protocol in automotive is LIN (Local Interconnect Network) [LIN, 2003]. It contains 16 master-slave nodes without a repeater. It is characterized by relatively simple frame structure and low bit rate, which benefits the investments.

By contract, fieldbus in locomotive is organized in a more hierarchical way. TCN (Train Communication Network) is applied in numerous up-to-date train control systems, such as ICE (Intercity-Express) 3 in German railways and IC 2000 in Swiss federal railways. It consists of two fieldbus systems: MVB (Multifunction Vehicle Bus) among equipments aboard each train section [IEC-61375:3-1, 2015] and WTB (Wire Train Bus) of connecting them together [IEC-61375:2-1, 2015]. The data rate of MVB is 1.5 Mbit/s data rate while WTB is 1 Mbit/s.

As for the application field of manufacturing, Profibus (Process Field Bus), with its advantages such as cheap coupling and integer bit formats, has gained large support in Germany. It is based on differential bus hardware with a bit rate up to 12 Mbit/s. Furthermore, more lights are currently shed on PROFINET (Process Field Net) and Ethernet because of their higher bit rate, multiple interface solutions and efficiency in performance [Felser, 2005].

Another application of fieldbus is aeronautics for purposes of both military and civil [Munoz-Castaner et al., 2011]. ECUs (Electronic Control Units) in America are standardized with the 1553b fieldbus protocol [Burton et al., 1988].

Increased focus has been brought in the field of building automation. BACnet (Building Automation and Control Network) is used as networking the applications, such as HVAC, lighting, access and fire alarm systems [ASHRAE, 2005]. In Europe, KNX (Konnex), as a fieldbus-based protocol for building automation, is standardized by EN 50090 [EN50090, 2005].

To sum up, one fieldbus type, such as Ethernet, cannot meet various applications with different requirements and interests. Considering Industry 4.0, one or several fieldbus protocols may have been shared as mainstream for their capabilities of efficient communication and adapted connectivity. Nevertheless, the application differences would still be distinguished among different competitors in the long term, resulting in remaining the customer-binding-oriented diversity among fieldbus protocols and their systems.

	Application fields						Max. node per segment	Max. segment length (m)	Max. bit rate (Bit/s)	Investment price level
	Automotive	Manufacturing	Process control	Avionics	Locomotive	Building automation				
CAN	X	X					32	40	1M	Medium
Profibus		X	X		X		126	120	12M	High
Ethernet	X	X	X			X	100	100	10~100M	High
LIN	X						16	40	20k	Low
1553b				X			32	30	1M	High
MVB					X		255*	2000**	1.5M	High
WTB					X		32	860	1M	High
KNX						X	64	700	9.6k	Low
BACnet						X	127	1000	10~100M	High
SmallCAN						X	1000	1000	9.6k	Low

Table 2.1: **Review of fieldbus types and their quantities related to this work**
Note:
* 255 configurable and programmable stations or up to 4095 sensors/actuators.
** 2000 meter using OGF (Optical Glass Fibers); 200 meter using shielded twisted pair with RS 485; 20 meter using ESD (Electrical Short Distance) without galvanic isolation.

Based on this, requirements of different aspects can be generated in Table 2.1.

Requirements of fieldbus structure:

- Fieldbus extension
    1. max. 100m in manufacturing
    2. 40m in machine or in automotive vehicle
    3. max. 1000m in building automation
- Bus nodes per segment

1. 16, 32, max. 100 nodes in automotive
2. max. 126 in manufacture
3. max. 255 nodes in locomotive
4. max. 1000 nodes in building automation

- Bit rate
  1. 20k to 1M, in future max. 100M in automotive
  2. 1M to 12M in manufacture
  3. 1M, 1.5M to 12M in locomotive
  4. 9.6k, with Ethernet then max. 100M in building automation

Concerning the application field of building automation, complexity has a great influence on system performability. Therefore, the complexity generated from the fieldbus structure need to be profiled and analyzed by applying suitable description means and methods.

2.2 Description Means and Their Adaptions

2.2.1 UML Class Diagram

The UML (Unified Modeling Language) Class Diagram is introduced in this section as one description medium in this formal system approach for visualization, specification, construction and documentation in various concept levels and system levels. The expression of the UMLCD is comprehensive enough to learn and adopt in this work.

UMLCD, a subset of the whole standard modeling language, is used to map the system design, which provides vocabulary and the rules focus on the conceptual and physical representation of a system. The UMLCD consists of three major elements: the UML's elementary building blocks, the rules that dictate how these building blocks can be bound, and some common mechanisms that apply throughout the language [Booch et al., 1999]. UMLCD emphasizes the structural relationships among the interactive objects. Therefore, it is appropriate for describing and characterizing the structure and behavior of Fieldbus system in particular.

Building Blocks: the terminology consists of three sorts of elementary building blocks:

1. Things
2. Relations
3. Diagrams

Things are the abstractions that are primary elements in a model; relationships play roles of binding things together; diagrams configure selected collections of things.

Things that are properly depicted graphically are implemented in the UMLCD. Moreover, things that are properly illustrated textually are also realized in programming language. There are four kinds of things in the UMLCD: structural things, behavioral things, grouping things and annotational things.

Relations: Four kinds of relationships in the UMLCD, which are integrated for this work, are introduced as follows:

Definition 2.2.1. Association

Association is a bidirectional relationship, specifying that objects of one thing are connected to objects of another [Booch et al., 1999]. Association denotes a structural path, by which the bounding objects of the classes interact with each other. The basic graphic rendering for an association is a solid line. Association, as a general relationship, depicts the peer-to-peer classes. Moreover, multiplicity, aggregation and composition are deduced from association.

In many system characterizing and modeling situations, it is important to demonstrate the number of objects connected across an association. The number is then called the multiplicity of an association [Booch et al., 1999]. It expresses as numeric value that evaluates and specifies the model relationship. As mentioned in the bottom right Figure 2.1, the multiplicity can be identified as one 1, zero or one 0...1, one or more 1..$*$, or variable $*$.

Definition 2.2.2. Aggregation

Aggregation is a special sort of association, it is focused on indicating the independent structural relationship of “system and component”. It illustrates as a “has-a” relationship, showing that an object of the system has objects of its components. On the contrary, a plain association is a peer-to-peer relationship [Booch et al., 1999].

Definition 2.2.3. Composition

Composition belongs to aggregation, with forceful partnership and concurrency as part of the whole. Parts with uncertain multiplicity may be established after the self-composition process. It is worthwhile to be mentioned that the parts would always exists with the whole once created. These parts are also removable before the ending phase of composition.

The difference between aggregation and composition is that the former can live independently, whereas the part of the latter relationship cannot really exist without the whole. The graphical difference between these two aggregation relationships is that the shared aggregation is represented by adding a plain association with an open diamond at the whole end while the composition aggregation has a solid diamond at the whole end.

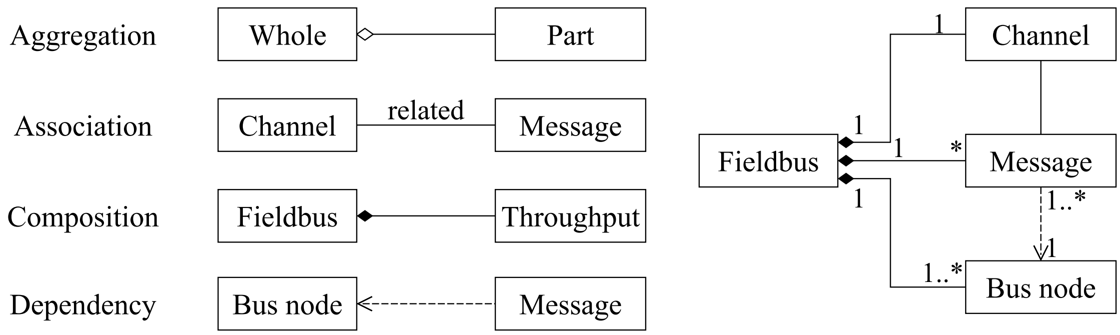


Figure 2.1: **UMLCD examples including aggregation, composition, association and dependency**

Definition 2.2.4. Dependency

Dependency is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing), but not necessary vice versa [Booch et al., 1999]. Graphically, a dependency is depicted as a dashed line. Directions are draw with the arrow pointing from a dependent to a provider. Dependency, one of the most common kinds of relationships, connects between classes that only use another class as a parameter to an operation. Create a dependency pointing from the class with the operation to the class used as a parameter in the operation. Dependency, as a form of association, is a relatively weaker relationship.

Diagrams are the graphical expressions of a set of elements, in most cases rendered as a connected graph structure of things and relationships. Class diagram are constantly used diagrams found in describing an object-oriented system modeling concept. A class diagram denotes a set of classes, interfaces and their relationships. It is essential for visualization, specification and documentation of the structural models. Moreover, it is also available in constructing executable systems through forward and reverse engineering [Booch et al., 1999].

2.2.2 OSI Model

The OSI (Open Systems Interconnection) model is a conceptual model. The purpose of selecting this conceptual model in this work is to describe the interrelations and feasibilities of the complex fieldbus system with its requirements. The fieldbus system can be divided into seven abstracted layers, which are listed as follows:

1. Physical Layer: transferring the logic concepts into physical specified signals, defining the communication topology.
2. Data Link Layer: providing the frame structure and the Medium Access Control (MAC), authenticating the frame transmission in a shared communication channel.
3. Network Layer: allocating the frame with assigned the information and the address to the transmitting designation.
4. Transport Layer: multiplexing and retransmitting solutions in fieldbus system, a case study is the TCP (Transmission Control Protocol) of the IP (Internet Protocol).
5. Session Layer:establishing recommissions and recoveries, providing half-duplex communication type in fieldbus systems.
6. Presentation Layer: transforming and mapping the data from the lower layers into syntactical and semantical concepts.
7. Application Layer: defining the functions and their interactions based software between the user and the fieldbus system, generating the data to transmit.

The applied access mechanisms in MAC are focused and described regarding Data Link Layer, seen Section 3.1.1. In addition, this work also covers the lasting message sequence in Data Link Layer and its functional mapping in Application Layer, seen Section 3.1.2. In order to proceed the validation procedure with the aspect of busload, the Bus channel also need to be concreted as a subsystem, seen Section 3.1.3.

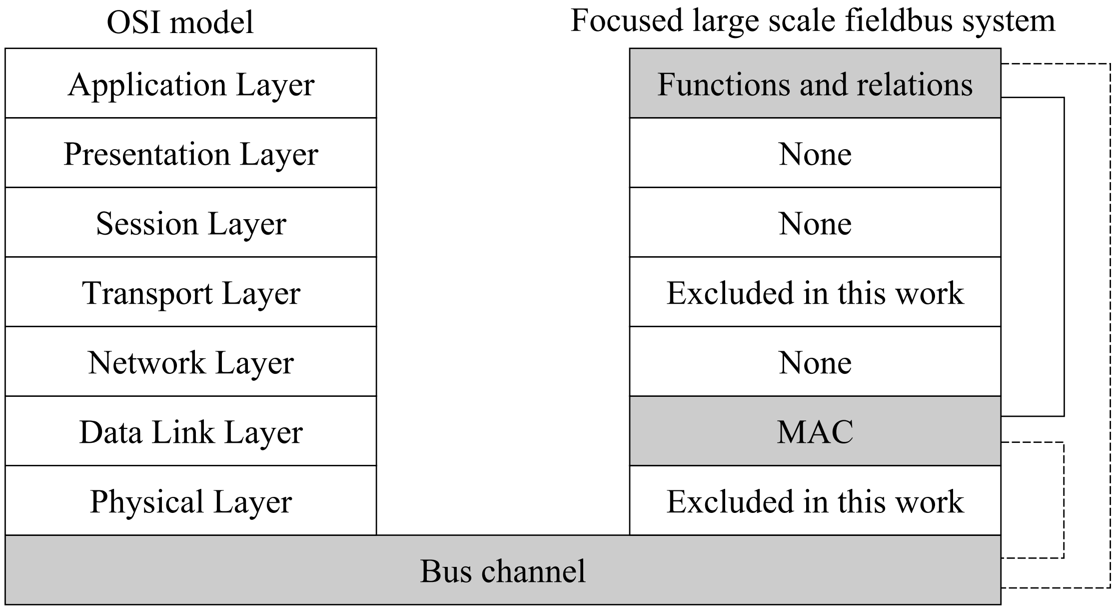


Figure 2.2: Mapping between OSI model and the focused fieldbus subsystem (grey)

2.2.3 Motivation and Definitions of Applied Petri nets

Under the backgrounding of complex fieldbus systems, methods of steady state analysis are enumerated as: Markov Chains [DIN61165, 2007], Queuing theory and Petri nets [IEC62551, 2012].

As is shown in Table 2.2, discrete-time Markov Chains, Continuous-time Markov Chains and SPNs are the mathematical models in research field of reliability. The transformation rate of a Markov Chain can be interpreted as the transition firing rate of a Petri net model. During the development process of a complex system, changes are often made in a iterative way. However, only the global system state can be modeled by Markov Chain [Schnieder and Schnieder, 2013].

On one hand, the traffic of fieldbus communication is dependable with interactivity and implemented access methods, the undetermined behaviors from which are complicated to reveal and profile. Concurrent sending, on the other hand, is a behavior of a communication system in which several events are simultaneously executing and interacting with each other. Petri net is a mathematical model developed also for general concurrent computation, in order to support reasoning, specification, testing and simulation. Therefore, Petri net is mainly adopted in this work to model and analyze the system complexity. This is because of its graphical and mathematical competences in describing concurrent systems [Ehrig, 2003].

Table 2.2: Review of the relevant description means

Category	Characteristic	Markov Chains	Queuing theory	Petri nets	DSPN model
Design phase	Operation		X	X	
	Implementation			X	
	Modeling	X	X	X	X
	Specification			X	X
Complexity	Modularity			X	X
	Time mapping	L	X	X	X
	Parallel process		X	X	X
	Sequential process	X	L	X	X
Validation	Complex system	L	X	X	X
	Available tools	L	L	X	X
	Analysis oriented	L	X	X	X
	Simulation oriented	X	L	X	X
Pragmatic	Causality oriented			X	
	Method oriented			X	X
	State oriented	X		X	
	Structure oriented		L	X	X
	Stochastic distributions	L	L	X	X
	Mathematical	X	X	X	X
Paradigmatic	System oriented			X	X
	Behavior oriented		X	X	X
	Object oriented				
	Function oriented				X
	Data oriented				X

Note:
For analyzing the characteristics, X means it is capable while L means it is weakly adequate. This review is based on the works of [Schnieder and Schnieder, 2013] and [Schnieder, 2013]

Acoording to the work of [Schnieder, 2013], Petri nets are intended to visualize the dynamics of quantified system. It has been introduced in [Carl Adam Petri, 1962]. The descriptive power of Petri net applied in profiling and modeling the continuous and discrete Systems is mentioned in the works of [Chouikha and Schnieder, 1998] [German, 2000] and [Zhu and Schnieder, 2000]. Moreover, [Fay and Schnieder, 1999] and [Zhu and Schnieder, 2001] present the timed analysis with Petri net model-based simulation. The complexity of functionality of train control system has been modeled by Petri net, as mentioned in [zu Hörste et al., 2013].

The initial purpose of this theory was invented to describe the concurrent transmission behavior of a communication system in a homogeneous and exact way. In fieldbus system, message flow is ascribed to all nodes connected on the bus competing with each other to occupy the bus channel. In order to support system reasoning, specification and simulation, Petri net as a mean of description can be used to describe and characterize the emergent properties of the complex fieldbus system. In this work, Petri net is selected to profile the message-sending concurrency and the message collision.

According to work of [Schnieder and Schnieder, 2013], the advantages of applying Petri nets are listed as follows.

1. Petri nets can be applied in comprehensive modeling-based system abstraction levels, from model concept to specified scenarios.
2. mapping abilities of timed behavior, sequencing and redundancy.
3. transitions are available for parameterizing one or more stochastic distribution type.
4. Petri nets are premised on a precise mathematical background. Therefore, requirements are fulfilled by a specific property analysis.
5. Hierarchical structure of the system and its model concept can be profiled by Petri nets.
6. Relatively flexible changes during system design phase.

In real fieldbus systems, the message-sending frequency resulting from the interactions between the system and is environment might have stochastic characteristics. Therefore, the selected Petri net method is introduced step by step. Petri net is generally presented in Definition 2.2.6, and DSPN as extensions of Petri net is mainly adopted for timed analysis of traffic behavior, which is introduced in Definition 2.2.7.

Definition 2.2.5. Net [Mueller, J. R. and Schnieder, E., 2008]

A net is formally defined as a triple $\Sigma\ N = (S, T, F)$ where,

- S is a set of states
- T is a set of transitions, disjoint from S ($S \cap T \neq \emptyset$)
- F is a states- and transition-binding relation $F \subseteq (S \times T) \cup (T \times S) \to \mathbb{N}$

the binding relation denotes the arc-based sets, which pre-state will be taken over by the post-state. In this work, states symbol S is replaced by P, inherited from the Petri net concept.

Definition 2.2.6. Petri net

Petri net is defined as a 7-tuple $\Sigma\ PN = (P, T, I, O, C, W, M)$ where,

- $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places.
- $T = \{t_1, t_2, \cdots, t_m\}$ is a finite set of transitions, and $P \cap T = \emptyset$ and $P \cup T = \emptyset$.
- $I \subseteq (\mathbb{N}_0^n \to \mathbb{N}_0)^{n \times m}$ is a matrix of the marking-relevant input arc set. The $(i, j)th$ entry of the matrix gives the possibly marking-relevant arc set of the input arc from p_i to t_j. $\mathbb{N}_0^n \to \mathbb{N}_0$ is the set of functions from $\mathbb{N}_0^n \to \mathbb{N}_0$.
- $O \subseteq T \times P$ is the set of output arcs, contrary to the input arcs I
- $C \to \mathbb{N} \cup \{\infty\}$ is the arranged capacity of each place.
- $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is the weight function, indicating that each arc is assigned with a non-negative integer.
- $M = \{M(p_1), M(p_2), ..., M(p_m)\} : M_0 \subseteq \mathbb{N}_0^n, P \to \{1, 2, 3, ...\}$ is the initial marking, M can be seen as a vector given by $M_k = \{M_1, M_2, ..., M_i, ..., M_n\}$ where the i^{th} entry of M is M_i, which is also the marking of the place p_i. $M(p_i)$ is the number of tokens in place p_i; it represents the tokens assigned in every place element of the net in the initial phase.

Figure 2.3: **Basic elements of Petri net**

Token, graphically shown as a black dot shown in Figure 2.3, occurs only in the places. It shows the number of available resources, an enabled condition or an ongoing process. Due to actual constrains, maximum limit number of tokens C is assigned with each place.

The relation of a Petri net definition between a four-tuple $PN = (P, T, F, W)$ and a four-tuple $PN = (P, T, I, O)$ is that $F \subseteq (P \times T) \cup (T \times P)$ represents the flow relationship, $W : F \rightarrow \mathbb{N}^+$ is a weight function, N^+ is natural number. Weight is defined as follows: if $I(p_i, t_j) = k$, where $k \geq 1$ is an integer, a directed arc from place p_i to transition t_j is labeled with weight k. For example, if $k = 1$, an arc without label is drawn between the corresponding P and T; if $k > 1, k \in \mathbb{N}$, a directed and labeled arc with its multiplicity is bound between its P and T; however, if $k = 0$, then no arc exists between the relevant P and T.

$I \wedge O$ compose of the net flow relation F, which denotes all the associated relations between the whole set of places and the whole set of transitions $F \subseteq (P \times T) \cup (T \times P)$. There are no direct links between places and places or between transitions and transitions, *i.e.* $F \subseteq (P \times P) = \emptyset$ and $F \subseteq (T \times T) = \emptyset$.

A Petri net with the proposed initial marking is symbolized by M_0. Any enabled firing transition would generate a new marking M': For $\forall p_i \in P$, if $M(p_i) = m$, which denotes that the number of m tokens is assigned with place p_i.

In mathematical descriptions, transition $t \in T$ is enabled, if and only if $M(p) \geq I(p, t)$; $\forall p \in P$. The equation describing the marking transformation is shown as follows:

$$M'(p) = M(p) - I(p, t) + O(p, t); \forall p \in P \tag{2.1}$$

As is depicted by the equation, transition firing consists of two steps: consuming tokens from all input places and generating tokens in the output places, which are represented by $-I(p, t)$ and $+O(p, t)$ respectively.

According to the transition firing rule, ${}^\bullet t$ denotes the preset of a transition t is the set of all input places of the transition t. On the contrary, $t^\bullet$ means the post set of a transition t is defined as the set of all output places of the transition t. Therefore, $\forall t \in T$,

$$\begin{aligned} {}^\bullet t &= \{p : p \in P\} \text{ and } I(p, t) \neq 0 \\ t^\bullet &= \{p : p \in P\} \text{ and } O(p, t) \neq 0 \end{aligned}$$

Similarly, ${}^\bullet p$ represents the preset of place p is the set of all input transitions of the place p. And $t^\bullet$ expresses the post-set of place p is the set of all output transitions of the place p. For $\forall p \in P$,

$$\begin{aligned} {}^\bullet p &= \{t : t \in T\} \text{ and } O(p, t) \neq 0 \\ p^\bullet &= \{t : t \in T\} \text{ and } I(p, t) \neq 0 \end{aligned}$$

With the adequate knowledge mentioned above, The transition $t \in T$ is enabled under a marking m can be represented as $m[t >$, if

$$\forall p \in^{\bullet} t : m(p) \geq 1 \tag{2.2}$$

The marking m transferring to new marking m', resulting from the transition t, can be expressed as $m[t > m'$ with

$$m'(p) = \begin{cases} m(p) & ifp \in^{\bullet} t \cap t^{\bullet} \\ m(p) - 1 & ifp \in^{\bullet} t / \ t^{\bullet} \\ m(p) + 1 & ifp \in t^{\bullet} / \ ^{\bullet}t \end{cases} \tag{2.3}$$

According to [Mueller, J. R. and Schnieder, E., 2008], the reachability of a Petri net is defined as follows.

Let a Petri net denoted as $PN = (P, T, F, m_0)$, if a transition firing sequence $\sigma = t_1, t_2, ..., t_n$ with $m_0[t_1 >_N m_1[t_2 >_N m_2 ... m_{n-1}[t_n >_N m_n$, then marking m_n is reachable in N from the initial marking m_0 due to the transition firing sequence σ, symbolizing as $m_0[\sigma > m$. The set of all reachable markings, formulated as $[m_0$, can be expressed by Equation 2.4.

$$[m_0 >_N := \{m | \exists \sigma \in T^{\bullet} : m_0[\sigma > m\} \tag{2.4}$$

Petri nets can also be classified as low-level Petri nets and high-level Petri nets. Low-level Petri nets, such as Stochastic Petri net, are principally applied in modeling, analyzing and validating the concurrent events. In low-level Petri nets, token represents the resource location of the local system. The token itself cannot be assigned with token types. On the contrary, each token in high-level Petri net encompasses complex data sets, characterizing the process state. The complexity of large scale fieldbus system is hereby hierarchical depicted and constructed by high-level Petri nets due to its advantage of structuring modules [Jensen, 1983].

Definition 2.2.7. Stochastic Petri net

According to [Mueller, J. R. and Schnieder, E., 2008], a Stochastic Petri Net (SPN) is a quintuple $N = (P, T, sT, F, \Pi, m_0)$ with:

- (P, T, F, m_0) is a PN, where transitions are divided into instant-firing transitions and timed transitions. Instant-firing transitions are depicted as thin bars and timed transitions as thick bars.

- sT is the set of stochastic transitions. They are graphically drawn either as white bars (exponential distribution type) or as gray bars (other stochastic distribution types, such as Gaussian, log-normal and uniform).
- $sT \rightarrow Prob$ is the probability function, with $sT \subseteq T$ is called the set of probabilistic transitions and $Prob := \{p \rightarrow \mathbb{R}^+ | 0 \leq p \leq 1\}$ is the set of probabilities. So, $\Pi(t)$ specifies the firing probability of transition $t \in sT$, with $(^\bullet t)^\bullet = \{t\} \implies \Pi(t) = 1$.

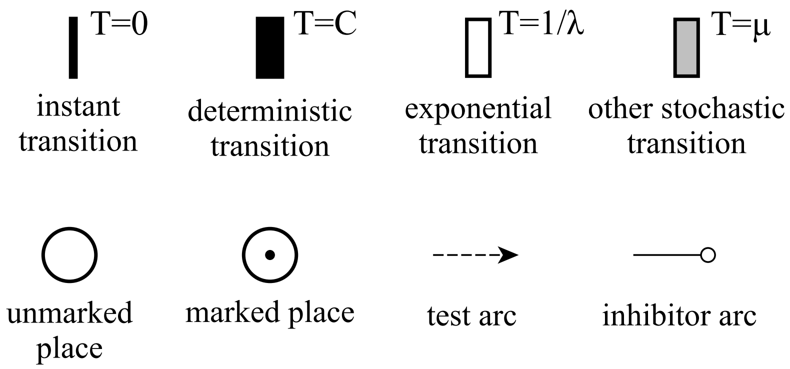


Figure 2.4: SPN graphical elements

In this work, transition types shown in Figure 2.4 are differentiated by the firing time interval T in each transition type.

1. $T = 0$, the transition immediately fires when enabling.
2. $T = Constant$, the transition fires with a deterministic time delay when enabling.
3. $T = 1/\lambda$, where λ represents the expectation of the exponential distribution type.
4. $T = \mu$, where μ represents the expectation of the other distribution type, such as the lognormal distribution.

Places, representing the system state, can be categorized into two classes: marked and unmarked places. The latter indicates the system is ongoing the current state. The test arc do not consume the token from the pre-place, whereas the inhibitor arc enables the post transition when the pre-place is unmarked. These two types of arcs are hereby selected to map some conditional firing structure with system resources or context involved.

2.2.4 Applied Definitions in Probability Theory

The interactiveness between the fieldbus system and its environment is one of the dynamic characteristics leads to the system complexity. These quantities may be deterministic and stochastic. Therefore, probability theory can be used as an analytical solution to proceed performance[Lindemann, 1992]. As a result, stochastic message-sending characteristics can be quantified and later parameterized into the Petri net communication model.

2.2.4.1 Definitions regarding Probability Theory

Probability denotes a value of measurement, estimating the similarities that something is about to happen or whether an argumentation is right. Probability value is ranging from 0 to 1, indicating the likeliness from one thing that will never occur to one thing that will definitely occur. Probability is also estimating the system behavior with the number of occurring times of observed sample events.

Definition 2.2.8. Random Variable [Morris, 1986]

Considering a random experiment with sample space S, A denotes a random variable, X represents a function that assigns a real value to each outcome in S. For any set of real numbers A, the probability that X would assume a value, containing in the set A is equal to the probability that the outcome of the experiment is also contained in $X^{-1}(A)$., which is defined in Equation 2.5.

$$P\{X \in A\} = P\{X^{-1}(A)\} \tag{2.5}$$

where $X^{-1}(A)$ is the event consisting of all points $s \in S$ such that $X(s) \in A$. A random variable X is said to be discrete if its set of possible values is countable. For example, each temperature message-sending event in the fieldbus system is regarded as a random variable A.

Definition 2.2.9. Stochastic Distribution of a Random Variable [Ross, 1996]

If a probability distribution is specified in the sample space S of an experiment, a probability distribution for the possible values of any random variable X can be fixed. A is any subset of the real line. Denote $P(X \in A)$ the probability that the value of X belongs to subset A. Then $P(X \in A)$ is equal to the probability that outcomes s of the experiment will be such that $X(s) \in A$. This can be defined by Equation 2.6.

$$P(X \in A) = P\{x : X(s) \in A\} \tag{2.6}$$

Definition 2.2.10. Distribution Function [Morris, 1986]

The distribution function F of the discrete random variable X is defined for any real number x by Equation 2.7.

$$F(x) = \sum_{y \leqslant x} P(X = y) = P(X \leqslant x), \quad -\infty < x < \infty \tag{2.7}$$

It is worthwhile to mention $F(x) \leqslant 1$. In reliability theory, $F(x)$ is adopted to describe the system probability failure: at the specific time x, $F(x)$ is describing the probability that the system is in fault state.

Definition 2.2.11. Probability Density Function [Morris, 1986]

A PDF (Probability Density Function) is a function that describes the relative likelihood for this random variable to take on a give value. It is the derivative of the distribution function $F(x)$ by its random variable x, shown in Equation 2.8.

$$f(x) = \frac{d}{dx} F(x) \tag{2.8}$$

Definition 2.2.12. Expectation [Morris, 1986]

The expectation or mean value of the discrete random variable X is represented by $E[X]$, shown in Equation 2.9.

$$E[X] = \sum_{x} x P\{X = x\} \tag{2.9}$$

Definition 2.2.13. Variance [Morris, 1986]

The variance of the random variable X is denoted by the Equation 2.10.

$$var(X) = E[X - E(X)^2] = E(X^2) - E^2(X) \tag{2.10}$$

Usually, the square root of $var(X)$ is named standard deviation, represented as $\sigma = \sqrt{var(X)}$

2.2.4.2 Stochastic Distribution

According to the applied stochastic distribution, the negative exponential distribution, normal distribution and log-normal distribution are mainly used and hereby introduced.

Definition 2.2.14. Negative Exponential Distribution [Simon, 2007]

The negative exponential distribution is a family of continuous probability distributions. It describes the time between events in a Poisson process. The probability density function $f(x)$ can be defined in Equation 2.11.

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geqslant 0 \\ 0 & x < 0 \end{cases} \tag{2.11}$$

Definition 2.2.15. Normal Distribution [Simon, 2007]

In probability theory, the normal distribution or Gaussian distribution is a very commonly occurring continuous probability distribution - a function that tells the probability of a number in some context falling between any two real numbers. The probability density function is defined by Equation 2.12.

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\dfrac{(x-\mu)^2}{2\sigma^2}} \tag{2.12}$$

Normally a normal distribution is symbolized as $X \sim N(\mu, \sigma)$. It is then named as normal distribution, if $\mu = 0$, $\sigma = 1$, and the normal distribution is symbolized as $X \sim N(0, 1)$.

Definition 2.2.16. Log-normal Distribution [Simon, 2007]

A Log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. If X is log-normally distributed, then lnX is normally distributed. The probability density function of Log-normal distribution is defined in Equation 2.13.

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\dfrac{1}{2}(\dfrac{lnx-\mu}{\sigma})^2} \tag{2.13}$$

2.3 Methodes for Modeling and Analysis

2.3.1 Model Concept and Abstraction Hierarchy

According to the works of [Schnieder, 2010] and [Yurdakul, 2016], a concept consists of three parts: a term, an expression and the hierarchical and contextual relations between the term and its expression.

The prerequisite of differing concept objects relies on the knowledge of distinguishing their properties and characteristics. Distinctive characteristic has a great influence on distinguishing one concept from the other. Therefore, various relations among concepts can be differentiated. Hierarchical conceptual relation exists between superior and subordinate concepts. A conceptual order is mono-hierarchically if one generic concept exists for each one. On the contrary, a conceptual order is poly-hierarchical if some concepts have multiple generic concepts. According to the work of [Schnieder, E. and Schnieder, L., 2010], these are two main forms of hierarchical conceptual relations.

Systems are interpreted with an abstraction hierarchy. The system consists of a set of components. They can be further demounted into subordinate stationary components. The meaning of the superior ordinate concept includes the connotation of the subordinate concept. The subordinate term differs in at least one additional characteristic from the superior concept. One case study is the lexeme "*control system*" is the superior ordinate concept of the lexemes of the sensor system, data processing and actuator system. The lower components also show a certain characteristic complexity similar to the super ordinate system. Therefore, the perspective of the system concept is composed of one super ordinate system and one or more subordinate systems. The reference system can in turn be considered as a super ordinate system of its own subordinate systems. An arbitrary number of levels are built. The system in each level can be denoted as subsystem. The subsystem can be seen as the detailed formation of system model concept, then multiple subsystems can be carried out into a single super ordinate system by abstraction and aggregation. Therefore, the level of the system abstraction together with its elements is determined by the investment and research on the degree of the required system specification. In addition, self-similarity of the system model concept is determined by the equal level of defined system properties.

The system concept is an interdisciplinary and explanatory model. It is applied in the fields of natural science (neurobiology), the engineering (System control theory and communication engineering), economics (business management) and social sciences. According to the work of [Schnieder, E. and Schnieder, L., 2010], the system model concept can be seen as one of the essential and integrative factors in further increasing branches of interdisciplinary research.

2.3.2 System Context and System Properties

According to the system concept defined by etymology, one system is rather detached from the concrete realizations, than the composition of elements, requisites and external lasting influences. The goal is to fulfill the specific purposes, such as system description and validation. The purpose can be explained as complex contextual influence with theoretical system models under describable specific circumstances, such as fieldbus system with its environmental interactions. As a result, description means influence on the acceptability and validity of the system models.

The system context, or named peripheral system, emphasizes on the interactive relationship between the system and its environment. The peripheral system is defined as the sum of all objects, the changes of which influence the system. On the contrary, some objects are modified by the behavior of the system (the impact of the system frame on the peripheral system). The peripheral system itself in turn comprises all characteristics of a system concept. Then, the system itself can be further identified as open system and close system. The difference of these two systems is determined by whether the exchange between the system with the peripheral system occurs. Fieldbus is traditionally recognized as a relatively closed communication system. According to the railway norm of communication safety [EN50129, 2003], fieldbus is identified as closed system compared to wireless communication. However, fieldbus-based building automation system still interacts and ameliorates with their contexts in a frequent and dynamic manner. Therefore, it is necessary to analyze such variances when shifting the same term even among same research focus with different application fields.

The system is with four fundamental properties: state, structure, function and behavior [Schnieder, 2010]. These four properties can be precisely refined in characteristics and quantities with a variety of relationships. The fieldbus system's properties are generally shown in Figure 2.5. It is then specified in Table 2.3.

The system has a **state**. The current state of a system can be clearly described by a set of state quantities by specifying the values of all constant or variable attributes. They describe properties, characteristics and quantities at a given time and place with the material, energy and information conditions included. It is worthwhile to be mentioned that these conditions are independent from the system description method, whether or how they are interpreted. Additionally, based on a dynamic characterization of a transient or steady state, it is important to distinguish a global or local state, the state can be viewed as an input or initial state or within a storage

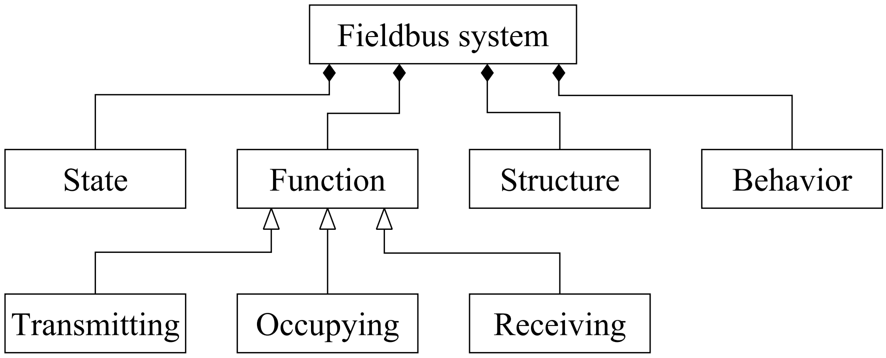


Figure 2.5: Properties of the fieldbus system describing with UMLCD

state at the system boundary. One example is the whole fieldbus system communication compared to the message generation inside each local node. The dynamic system behavior is defined by the assignment of a monotonic sequence states. It is expressed in a variety of forms, which in turn are characterized by special characteristic allocation, for example the busload validation by means of Monte-Carlo simulation.

The system has a **function**. The input variables of the state quantities are converted into the output variables in the state quantities (material, energy, information) as the system functions. The function is partially fulfilled by multiple sub-function interrelations (storage, transmission, processing of the fieldbus message), which are linked to a system. These subordinate function blocks do not fulfill individually the overall purpose.

The system has a **structure**. It consists of a quantity of parts, which are interrelated with each other as well as the environment. According to the definition of the term structure mentioned in IEV 351 [7], the decomposition principle can be differentiated with respect of the structure regarding the different abstraction levels (hierarchical levels) in the inter-systemic relations and intra-systemic relations.

The system has a **behavior**. Dynamic state changes determine the observed system behavior. It is described and characterized by a logical order (causality) and a temporary sequence (temporarily) of state transitions in accordance with the system functions. The behavior of a system is thus obtained from the assignment of different states over time.

In order to illustrate the status, function, structure and behavior as four system properties in detail. Isn order to specify the modeling concept in fieldbus systems, Table 2.3 presents a case study of requirements specification of general fieldbus systems. Properties and characteristics related to fieldbus are hereby listed.

Properties	State	Function	Structure	Behavior
Characteristics and quantities listed with (+)	Emergent degree +Local state +Global state	Tasks +Generation +Occupation +Sending +Transmission +Receiving	Environmental reference +Input influence +Output influence	Algorithm (Non-deterministic behavior) +Goodness of fit +Monte-Carlo Method +Firing rate analysis
	Resolution +Quantification		Resource reference +Allocation +Partitioning	
	System relevant +Input +Output	Mathmatical description +Linear / Non-linear +Continuous / Discrete	Topology +Fieldbus	Mathmatical description +Petri net methods +Sending Frequency +Overload equation +Norm +Transition firing rates +Limit value +Stochastic Distribution
	Target relevant +Idle +Busy state	Functionality +Overload / Normal +Intact / defect	Topology +Fieldbus	
	Behavior reference +Initialstate +Ending state	Transmission function +Message scheduling	Function relevant + Open system +Closed system	Analysis +Transient +Stationary
	Existence +State probability +Transmission probability		Mesh degree +Coupling +Complexity degree	

Table 2.3: Requirements specification of general fieldbus systems

2.3.3 Attribute Hierarchy

The system term is clarified by four properties, which are mentioned in Section 2.3.2. According to attribution degrees, the property can be specified into characteristic and quantity. This system abstraction is burdensome and only available where a system is formed concretely. Two forms of system concretization are hereby categorized: function system (such as control system) and resource system. The latter is the real carrier of the former. It is worth mentioning that the resource system inherits all the properties of the abstracted system model concept. It also contains the physical properties under system realization and implementation phases.

Properties represent the abstraction of perceptible states from reality. Properties, in a linguistic level, can be verbalized and hence set as terms provided by multi-linguistic model. In order to extend this argumentation precisely, the observable properties are empirically deduced from observed characteristics. Therefore, properties arise from the induction and abstractions characteristics.

Characteristic is a fundamental element for concept description and identification. Therefore, it is the central building block of constructing one terminology. Characteristic can be objectively determined. Hence it can be interpreted as precise and clarified property, by which the object is assigned with each characteristic as its appearance feature in non-verbal reality. One is composed of different characteristics. They can be identified from their distinctive characteristic values. Moreover, every characteristic is only assigned with one characteristic value. For this reason, these characteristic values need to be specified with sufficient precision. Therefore, in order to ascertain the characteristic value for a given characteristic carrier, a fundamental method is necessarily involved. For example, monitoring, testing, counting and measuring. These are methods categorizing and specifying characteristic values. Characteristics are self-determined terms, hence measuring (such as continuous characteristic) or counting (such as discrete characteristic) is available for determining characteristic.

Quantities, in special cases, are the general characteristic values. Quantities in physics refer to a class of physical phenomena or a class of physical properties, which can be assigned to concrete phenomena in numerical scale of measured values. Quantities can be generated by using well-defined experimental conditions. Establishing a physical quantity involves both the topological definition (equivalence- and order relation) and the metric definition (specifications on a scale form, zero point and unit). Due to the language limit, the ambiguity exists that there are no ordinal quantities, only ordinal characteristics in correspondence to determining a quantity to ratio scaled characteristics. Such a defined quantity is a part of a quantity system as a set of variables and consistent equations, by which quantities create relations

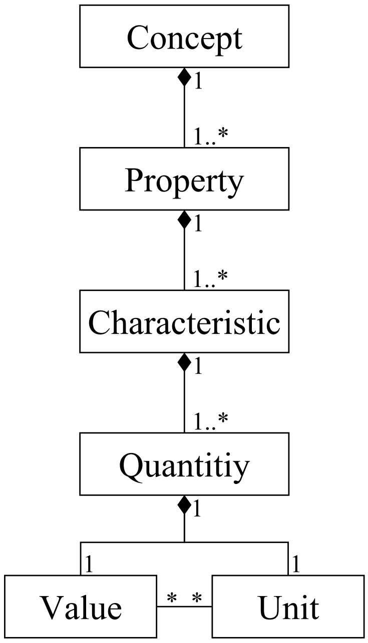


Figure 2.6: **Attribute hierarchy theory by UMLCD**

between them. Through embedding of a single quantity into a quantity system, a further quantity difference emerges between basic quantity and derived quantity.

Value and Unit, every special value of a quantity (quantity value) can be represented as the product of numerical value and its unit. The measurement unit mentioned here is a real scalar value defined by international standard, with which every other quantity value can be compared and expressed as a proportion of both quantity value and numerical value. Similarly, the differences also emerge between elementary units (meter is the basic unit of quantity length) and derived units (meter divided by second as derived unit of derived quantity speed).

2.3.4 Probability Distribution Fitting

In this work, analyzing the message-sending occurrence plays an important role in busload validation. The formal model is necessary to incorporate characteristics acquired and sorted from the real system. The mapping accuracy between the real complex fieldbus system and its model will be hereby improved. These characteristics, such as message-sending occurrence on the bus channel, might behave in a deterministic and stochastic way. Especially fieldbus systems in building automation, event-based message sending in this dissertation is mainly taken into consideration.

Considering the large scale fieldbus system in building automation, offices and floors with parallel ed function structures generate numerous event-based messages. They simultaneously request to occupy the bus channel. As a result, the busload of these messages in real system could behave as a burst. Therefore, a thorough quantification analysis of this stochastic characteristic improves the model quality and thus has a great influence on the model-based performance.

SmallCAN as a fieldbus system for building automation has been implemented in offices under the project "Future Workspace". Its server captures and records the channel' s traffic into the log file. Among these log data, the event-based message sending occurs quite often. A good case study is the temperature-message sending. They begin to simultaneously request to send on bus channel when environmental temperature varies.

The goal is to determine the probability distribution type of these event-based message-sending frequency. Therefore, analyzing the stochastic system behavior in real fieldbus system are introduced with following steps.

First, the event-based message-sending occurrences, such as temperature messages, in the log data is chronically selected and sorted. Second, the time intervals between every two sending occurrences of the selected messages are calculated. Third, these time intervals are further sorted into frequency with statistical data binning. Then, the binning is categorized with number of continuous sample length in order to form a histogram of selected message-sending with PDF as Y-axis.

The criterion of fitting the similarities between each fitting approach and sorted histogram is evaluated and determined by choosing the fitting approach with the highest likelihood value. The distribution fitting methods can be identified into parametric methods and regression methods. Regarding the context of this dissertation, the parametric methods are considered, mainly the maximum likelihood method is applied. This part will be further discussed in Section 4.6.

The results of the fitting approach, such as expectations and variances, can be parameterized into the Petri net model. The message-sending behavior of the model is consequently approaching to the real fieldbus system. The busload validation, such as Monte-Carlo simulation and transition firing analysis, can be henceforth proceeded.

2.3.5 Scenario Arts

With the help with the selected fitting method mentioned in Section 2.3.4, stochastic distribution types and their expectations of message-sending occurrence are fitted. As a result, message-sending scenario can be further modeled and parameterized.

Focusing on analyzing the concurrency of a large scale fieldbus system, message-sending scenario is necessary to be assigned with worst consequences occurred in an obtainable condition. Therefore, the worst-case scenario in this dissertation is selected. Its outcomes are to evaluate whether the emerged quality problems impacting on the performance of the complex fieldbus system are met with timing requirements, such as upper limit of channel throughput [Hansson et al., 2002].

The basic functional structure and are relatively similar among floors and offices in building systems. Therefore, analogical busload behaviors occur among these fieldbus-based building units. Especially, the throughput is accumulated by these event triggered message-sending occurrences.

A good case study is the temperature changing scenario caused by environment. The detected value of current temperature changes bigger than the predefined criterion in fieldbus system would generate a message onto the channel. This would occur in all offices and floors installed with temperature-related functions. As a matter of that, channel's busload would be severe if a considerable number of networked temperature messages spontaneously requests to occupy the bus system. This composes as a part of the worst-case message transmission scenario.

Worst-case scenario in this dissertation is hereby interpreted as the maximum networking topology of the Petri net-based message-sending models under a computerized condition. The evaluation work of worst-case scenario are specified in Section 6.3. By contrast, message-sending scenarios concerning average cases have been also modeled, simulated and evaluated respectively, see Section 6.2.

2.3.6 Computerized Methods for Model Extention and Simulation

According to Section 2.2.3, DSPN is applied as one of the formal description means for modeling the communication system. Based on this parameterized Petri net communication model, computerized model extension is hereby applied in approaching a complex fieldbus behavior in reality.

In building automation system, the complexity of fieldbus system can be interpreted as the message accumulation from numerous building unites. These building unites usually have similar spatial and functional structures among each floor in office building. Therefore, this Petri net communication model is hierarchically constructed with one bus channel state sub-model and variable number of message-source-nodes sub-models. They are further parameterized with the quantities abstracted and analyzed from the real fieldbus system.

Markov Chain Analysis is a sophisticate method for analyzing stochastic process description of complex systems. However, distribution type are limited in Markov Chain's regarding failure rate and repair rate of reliability [Gilks, 2005].

In this work, the timed transitions of Petri net model contain deterministic and stochastic parameters to describe complex fieldbus systems' behaviors. Besides, the stochastic distribution types of the model parametrization includes not only exponential, but also normal and uniform distributions. Therefore, classic Markov Chain method is not adequate for analyzing this hybrid stochastic behavior.

State space explosion problem in Markov methods are not inevitable, see the work of [Mahmoud, 2000]. It is another reason of applying Monte-Carlo Simulation instead of Markov Chain Analysis. In addition, , methods of performance and availability analysis need to be shed light on in the context of complex fieldbus systems in building automation.

Monte-Carlo Simulation is hereby used to generate the results of the DSPN model. Timed parameters need to be assigned to the DSPN model before starting the Monte-Carlo Simulation. Allowed stochastic parameters are expectations and variances of different stochastic distribution types, including exponential, normal, log-normal, Uniform and Weibull distributions. Each of these parameterized transition can be observed respectively.

In this work, the interactions and interrelations among functions in complex fieldbus system need to be described in a formal and precise way. However, the limited describing capabilities and approximate analysis method are another disadvantage of applying Markov Chain analysis and Queuing analysis [Bobbio et al., 2008].

In the research field of reliability, Queuing theory has been applied in temporary analysis. Queuing model can be equivalent to the SPN model [Schnieder, 2013]. However, the structural and emergent bus behavior need to be decomposed in complex fieldbus systems. In addition, log data-oriented analysis and parameterization are further discussed in Chapter 4. Therefore, Monte-Carlo method is selected for DSPN model simulation.

With the help of this computational method, transition firing rates can be generated. Each result of the observed transition has a confidence interval. The Monte-Carlo Simulation runs slower with the restricted confidence interval. In addition, upper limit of simulation steps can be assigned. Therefore, simulation process can be ceased with two criteria: fulfillment of the defined confidence interval and reaching of the maximum simulation steps.

In this work, this upper limit is set to one million steps. The maximum DSPN model is integrated with 200 subnets and 82044 net elements. Consequently, each simulation period has reached to twenty minutes to retrieve results of each observed transition.

2.4 Chapter Conclusion

First, an requirements overview of typical fieldbus systems with different application fields and performance-related parameters concerning this work is presented. Second, applied means of descriptions are illustrated, including UMLCD, OSI Model. Means of descriptions regarding formal modeling and quantitative analysis, such as the motivations and definitions of applied Petri nets and probability theory, are also discussed. Third, model concept and system abstraction hierarchy as system-related methods are presented. Definitions of system properties and system context are then explained. Then, methods regarding modeling and parameterization are gradually presented. As for the methods of model extension and simulation, applied computerized method and its motivation are given.

Chapter 3

Specification and Formalization of the Busload Validation

With the applied description means and methods in this dissertation, the focus of Chapter 3 is on formulating and formalizing the busload validation procedure.

The overview of SmallCAN system specification regarding busload is structured by the attribute hierarchy method. Second, these specifications are discussed regarding the Application Layer, Data Link Layer and bus channel subsystem of SmallCAN respectively.

The system concept, properties' categorization and its context in this dissertation are further specified into the validation procedure. This is concluded based on the mapping between UMLCD and DSPN. As a result, the interlinking between a communication system and its means of formal descriptions is specified.

Finally, the BMW principle and related projects are selected to structure the validation procedure focusing on the system complexity.

3.1 Specification regarding OSI Model and UMLCD

According to [Schnieder, 2010], the system can be represented by its subsystems and their relations. These can be structured by applying the attribute hierarchy method, which is mentioned in Section 2.3.3. The selected properties, characteristics, quantities, values and units are listed and further sorted as part of formal validation procedure.

In addition, with the help of the OSI model concept, specifications of SmallCAN system and its subsystems in this dissertation can be concluded and discussed within the Application Layer and Data Link Layer. They play a cause-and-effect role in generating the busload onto the channel and accumulating the complexity of the fieldbus system. Besides, bus channel is hereby considered as the third subsystem because of the focused busload concept in this work. They are hereby presented and selected for structuring the specification complexity of the SmallCAN system.

3.1.1 Application Layer of SmallCAN by Attribute Hierarchy

As mentioned in Section 2.2.2, Application Layer is in charge of the functions and their interactions defined by the software architecture, generating the data to transmit. It is the highest level in OSI model between fieldbus system and customer. With different SmallCAN applications, such as offices, laboratories or private house, characteristics and their further quantities vary.

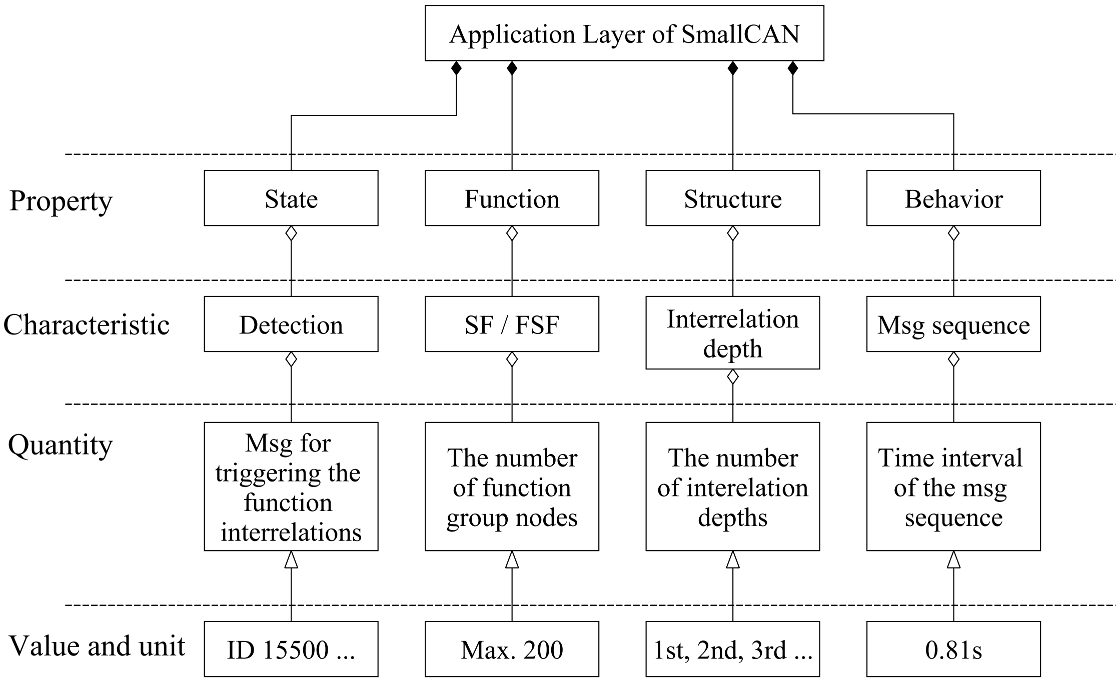


Figure 3.1: **Requirement specification of Application Layer of SmallCAN system**

Function: Figure 3.1 describes the selected specifications of Application Layer of SmallCAN system. The complexity in this subsystem can be described by the functions and their interactions. Definitions and categorizations of SmallCAN functions are categorized as follows.

- SF (Special Function) is defined to control the basic functions of the field drain devices, such as sensor and actuator control. The complexity specification regarding this function type is quantified as the large number of the accumulating SF nodes.

- FSF (Free-located Special Function) as its definition is not necessary to be installed in specific controllers. It is mainly in charge of executing the control structure by comparing and calculating the values between messages and the set-point, predefined inside this APL subsystem.

Structure in this case represents the function architecture. It consists of the number of SF / FSF and the number of their interconnections predefined in the APL of SmallCAN, generating the function sequence. The structures of function sequences vary among the different scenarios. Therefore, the structural and functional complexities differ on application configurations in building automation.

Behavior can be interpreted by the execution of the message sequences, which contain the function sequence defined in the structure. It is then quantified by the time interval of transmitting this message sequence.

State of the AP layer in this work focuses on the characteristic of detection. It is defined by detecting the first message of the function sequence predefined in this layer. The temperature control as a case study of the function sequence is hereby presented. Message ID 15500 contains the state information of the moving sensor, triggering the further function executions.

The complexity specification regarding this function type can be quantified as the number of functions and function interconnections. The function structure and message execution regarding this part are further discussed in Section 7.2.

3.1.2 Data Link Layer of SmallCAN by Attribute Hierarchy

According to the OSI model described in Section 2.2.2, Data Link Layer is one of the bottom layers (second lowest layer above Physical Layer). DLL contains two sublayers: LLC (Logical Link Control) sublayer and MAC (Medium Access Control) sublayer. MAC is mainly discussed in this dissertation. It is in charge of constructing the frame structure and authorizing the frame transmission onto a serial communication channel.

As is shown in Figure 3.2, the properties and characteristics in this layer has a great impact on the model-based validation procedure. Therefore, performance of applied access mechanisms defined in DLL need to be analyzed not only in the communication model but also in real fieldbus system.

State of DLL is characterized with concurrent and normal state due to the aspect of concurrent message-sending. Since the availability and function integrity in

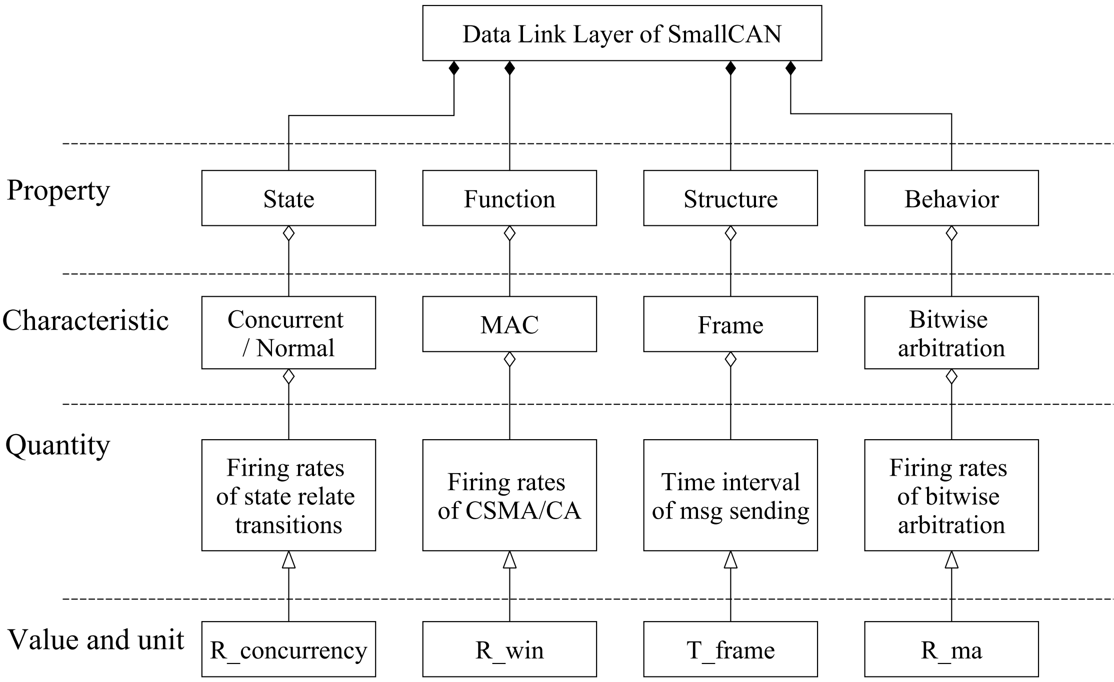


Figure 3.2: Requirement specification of Data Link Layer of SmallCAN system

SmallCAN is remained, the concurrent state is tolerable considering the serial communication topology of its fieldbus system. The availability analysis concerning this property is further discussed in Section 6.1.

Function is specified with access mechanisms in MAC. One CSMA/CA is selected and implemented in order to handling concurrent traffic in SmallCAN. CSMA/CA will be triggered when the concurrent message-sending scenario is detected. The CSMA mechanism together with its categorization is discussed in Section 4.4.

Structure is mainly represented by constructing one frame. It mainly contains a starter, a unique address ID, a required transmitting information and a CRC (Cyclic Redundancy Check). The structure comparison between one SmallCAN frame and one CAN frame is shown in Figure 4.2 in Section 4.2. With the timed aspect, this structural property is quantified by the time interval of one complete sending event.

Behavior in DLL means the process of handling the concurrent sending after triggering the CSMA/CA. This part is also described and quantified in Section 4.4.

After the quantification analysis, the CSMA/CA mechanism and Bitwise arbitration have been modeled and parameterized as part of the Petri net communication model, presented in Chapter 5. The evaluation of these integrated access mechanisms under different traffic density can be found in Section 6.2 and Section 6.3.

3.1.3 Bus Channel subsystem by Attribute Hierarchy

Bus channel in this work is defined as a shared logical connection instead of a physical transmission medium, such as a wire. In SmallCAN system, it handles the mapping results generated from APL and DLL.

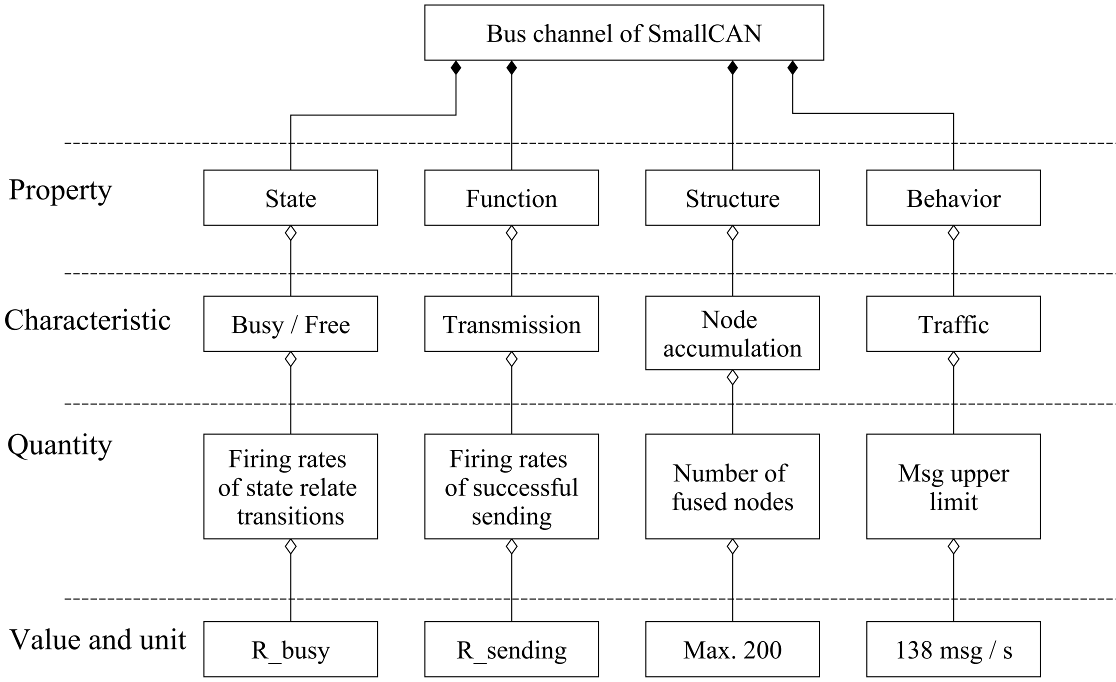


Figure 3.3: **Requirement specification of bus channel of SmallCAN system**

State of bus channel can be intuitively characterized as busy and free. The modeling and explanation of the bus channel state can be found in Section 5.1. The state is further quantified by the firing rates of the channel state related transitions.

Function of bus channel is in charge of message transmission. This can be quantified by the firing rates of successful transmission.

Structure is represented by the number of networked message-group-nodes onto one channel. The complexity arises when involving large number of nodes and their function interactions. Due to the current limitation of computer resource applied in extending the Petri net model, current supported networked message-group-nodes can reach up to 200. This quantity is set as the prerequisite of further model-rate-based analysis, shown in Chapter 6.

Behavior of bus channel is characterized by its message traffic generated from the subsystems mentioned above. Due to the busload validation procedure, the goal is to validate the transmitting capacity theoretically proposed [Schrom et al., 2011]. Therefore, it can be hereby interpreted as a third subsystem.

3.1.4 Specification of the SmallCAN System and its Context

As shown in Figure 3.4, the specifications among subsystems mentioned above are synthesized by attribute hierarchy with the aspect of busload, providing a formal validation procedure.

SmallCAN system is composed with networked nodes, channel and environment. Nodes can be represented with communication source / sink. Therefore, a message-sending occurrence is formed by transmitting and receiving messages among them.

Regarding the message-sending occurrence, interactions between the system and its environment have a great impact on generating the event-based traffic onto the fieldbus channel. Therefore, the quantitative analysis based on the such characteristics, such as the event-based message-sending occurrence, need to be proceeded by analyzing the message sending behavior in real system. Message-sending frequency can be quantified by fitting the stochastic distribution types, as is further shown in Section 4.6.

The busload is defined as the whole traffic on the channel. It is composed of stochastic message-sending occurrences, deterministic traffic and the message sequences of executing specific function sequences. The complexity of fieldbus systems is obviously a new characteristic in building automation. Any increase of these traffic types burdens the shared serial channel. In particular, the event-based message-sending occurrences contribute the most to this new characteristic. It can leads to a short-term message burst, which causes concurrent sending.

As a result, any concurrent-sending scenario will trigger the applied access mechanisms, such as CSMA/CA and bitwise arbitration. They are quantified by the relevant firing rates of the model-based analysis.

The emergent bus behavior can be characterized as overload and message burst. Overload can be quantified with the upper limit of the channel capacity, theoretically defined as 138 msg/s. One specific message burst is measure as 0.81 second. The detailed discussion concerning the messag burst can be found in Section 7.2.

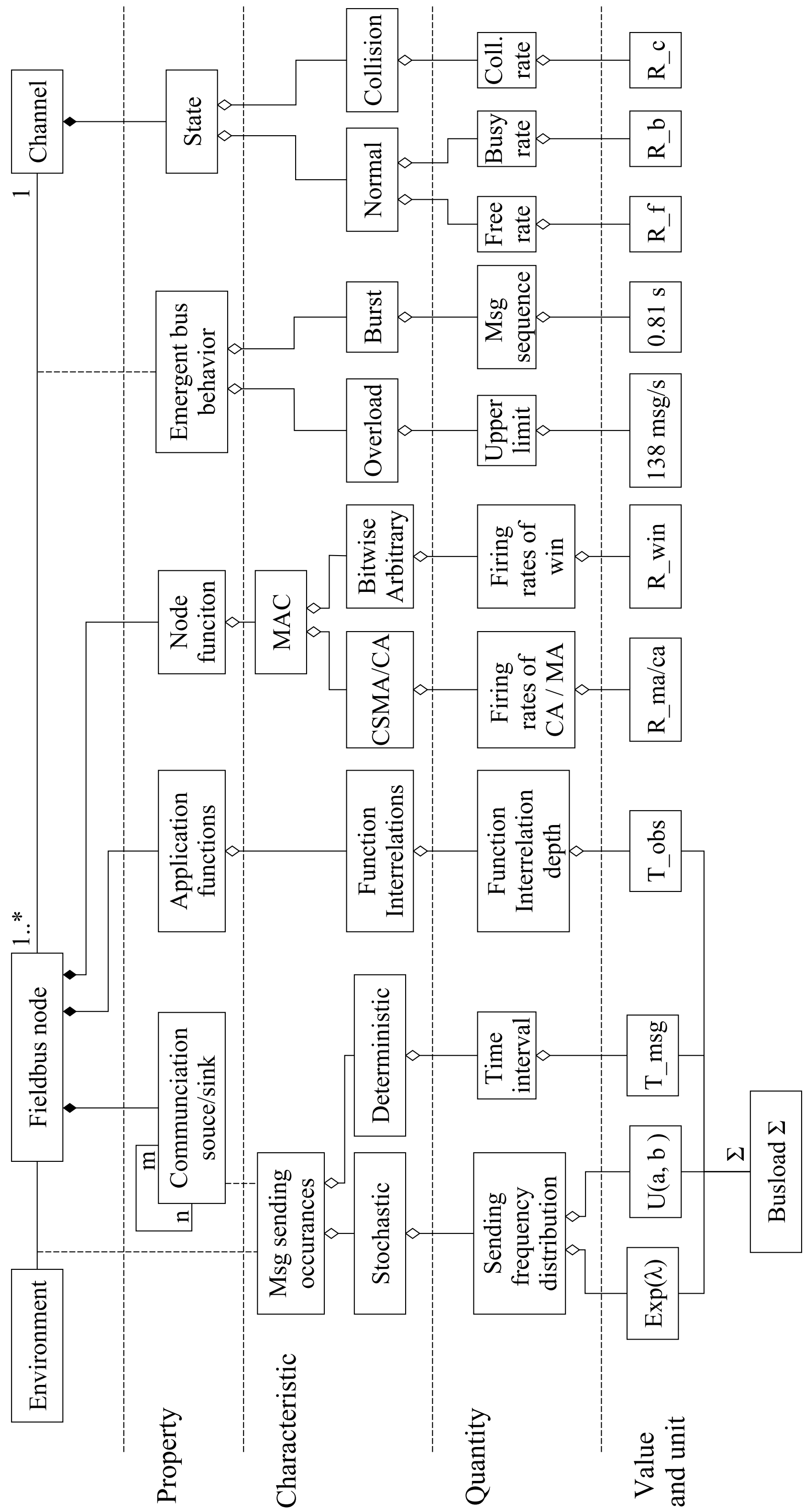


Figure 3.4: **Requirement specification of the whole system**

3.2 Elementary and Emergent System Properties

The system properties structured in Section 3.1 can be further sorted into categories: elementary property and emergent property.

Intuitively, elementary properties refer to those already existed properties maintaining inside the components which directly transfer themselves into the system properties without varying. Elementary properties are sustaining the whole systems fundamental functions and states. In the fieldbus system with large scale couplers, system resources such as hardware and wiring cables, are belonging to the elementary properties of the fieldbus system.

Elementary properties of a system are the states in the sense of existing physical and informational properties, characteristics and quantities (variables and values) within a given time. According to the mapping rule of the system state function, state quantities of input variables are converted into the state quantities of the output variables.

Definition 3.2.1. Emergence
According to the work of [Mueller, J. R. and Schnieder, E., 2008], a property of a system S is called an emergent property, if it occurs only by or can be explained through interactions of subsystems s that belong to S. An isolated view on the $s \in S$ won't explain the very property. Emergent properties constitute hierarchical levels: an emergent property ep of level l_k is being created by the interaction of subsystems on level l_{k-1}. It indicates that the interactions at the subordinate levels of complexity cause emergent properties on super ordinate levels.

An open and complex system formed by combining its subsystems. In reality it is formed by the interactions of the system properties. It is previously not observable and not explainable from the properties of its isolated subsystems. This phenomenon refers to as the emergence property according to system theory. Due to the work of [Schnieder, E. and Schnieder, L., 2010], the system thus can be described and identified with elementary and emergent properties.

Emergent properties describe system properties, in particular referred to the system behaviors, the new characteristic of the entire system. These properties are not attributed to elementary properties, but the specific selective elements combination (system structure-oriented) through the system states. The emergence corresponds closely with the recursivity of the system modeling concept. Different hierarchical levels can thus be interpreted as emergence levels. As for the SmallCAN system and its subsystems, this hierarchical levels are described in Figure 3.5.

The transition between the different emergent levels (or abstractive hierarchy levels) has been taken conceptually through the emergency relation. This relation can be

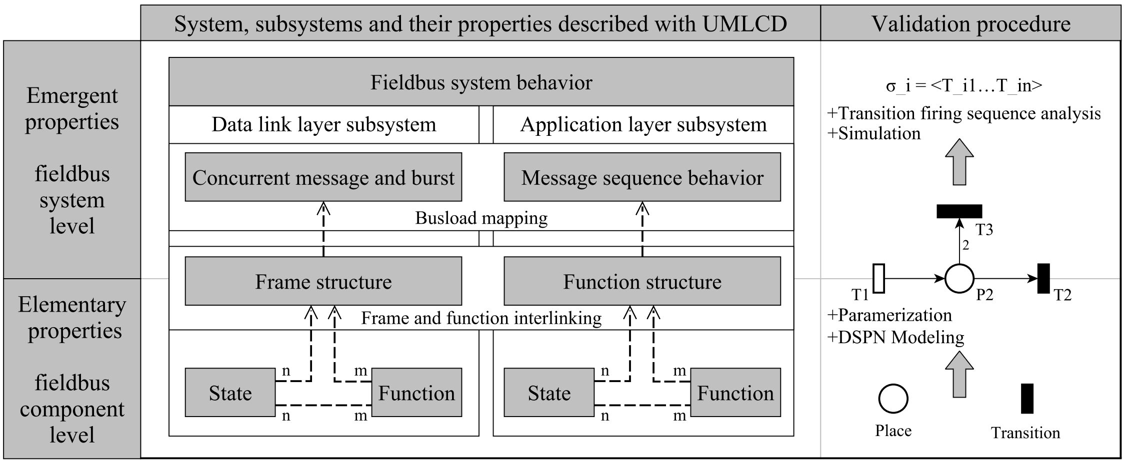


Figure 3.5: **Mapping of UMLCD and Petri net regarding validation procedure**

explained by the system structure, which presents itself as a deployment, or resource structure as well as a functional structure. Therefore, system behavior is interpreted as the result of these combinations. A process comprises the partial quantity of system changes from a certain initial or previous state or a sequential following state. The aspects of the deployment and resource structure, which determine the operational and functional structure, are discussed in the following section related to the concept model of functional and resource allocation.

An important emergent property for automation technology is dependability. It is guaranteed that a system stability ascribed by the highly reliable combination of two or more redundant emergent properties. For example, stability is produced through the emergence of forming function structure state with feedback in control systems. The availability also arises emergent for example by structural redundancy of the parallel subsystems and multiple selection, as well as maintenance activities of failed subsystems reflected in the functional state. Therefore, the more numbers of subsystem dimensioning the elementary or emergent properties, the more influence it results in the system dependability.

According to the Petri net theory mentioned in Section 2.2.6, the formal interpretation of elementary properties, such as states and functions in subsystems, can be specified with a set of places $P = \{p_1, p_2, ..., p_n\}$ and a set of transitions $T = \{t_1, t_2, ..., t_m\}$, the function binding, between different subsystem structure, can be also mapped with the token flow $F \subseteq (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$. As shown in Figure 3.5, the Petri net PN can be analyzed through the transition firing sequence or reachability graph, which are mapped as the fieldbus system behavior, denoted as the emergent property. This mapping approach is hereby designed to formulate and formalize the busload validation approach regarding the attribute hierarchy theory and Petri net modeling.

In this case, the large scale fieldbus system has the typical elementary properties such as the frame construction. Each section of the frame is constructed with the specification defined in the fieldbus protocol. Therefore, behavior results in a sequence of functions and states as the emergent aspect of system level. The message-sending concurrency and emergent bus behavior can be characterized as two emergent properties of the system complexity of the large scale fieldbus system.

3.3 BMW Principle regarding the Busload Validation

With the help of structuring and categorizing SmallCAN system and its subsystems, a model-based validation procedure can gradually created. Busload validation in this work is to verify whether the channel traffic generated from the Petri net communication model meets the specifications without harming the function integrity.

Due to an appropriate modeling of the concept's properties and characteristics, a complete and unambiguous predication can be performed [Schnieder, 2008]. For the purpose of profiling the formal validation process of the busload concept in the fieldbus system, appropriate description means need to be selected to structure the SmallCAN system and its subsystems. Moreover, they are applied in hierarchically addressing the preferred properties, characteristics and their quantities related to formalize the validation assignments of the busload concept with proper descriptive means. Then selected methods need to be correspondingly depicted for solving the described assignments respectively.

The BMW principle (abbreviation in German as "*Beschreibungsmittel, Methoden, Werkzeuge*" translated as "*means of description, methods for design and analysis, tools for supporting methods and descriptions*") [Schnieder, 2013]. The BMW principle is hereby selected for gradually structuring and formalizing the busload validation procedure, as shown in Table 3.1.

The validation procedure is illustrated step by step. Each step is presented with its applied description means, method and supporting tools. Moreover, each step is weighted by its role in the validation work. The relevant projects supporting this dissertation are listed and explained as notes. Finally, related chapters and sections of each validation step are listed.

The procedure of validating the busload can be concluded into following steps:

Table 3.1: **Procedure of the busload validation by the BMW principle, including performance roles, supporting projects and corresponding chapters**

Validation procedure	BMW concept			Roles in validation						Related project	Related chapter / section
	Means of description	Method	Tool	Conceptual analysis	Parameterization	Modeling & extension	Simulation	Structural analysis	Temporal analysis		
Model concept	Language	System abstration	Text	x						SmallCAN*	2.3.1
Formulation	OSI Model	Attribute hierarchy	Visio	x	x					SmallCAN	2 & 3
Formalization	UMLCD	BMW principle	Visio	x						SmallCAN	2 & 3
Msg sending behavior	Log data	Goodness of fit	Matlab		x			x	x	SmallCAN	4.6
MAC	Diagram	Concurrent analysis	Visio		x			x	x	SmallCAN	4.4
Modeling	DSPN	Hierarchical model	π-Tool			x				EnEff:Stadt**	5
Timed assignments	Parameter	Parameterization	π-Tool		x				x	DIGAFLEX***	5 & 6
Model extension	PNML	Worst-case scenario	JAVA			x		x	x	DIGAFLEX	5 & 6
Generating firing rates	T_rate	Monte-Carlo	π-Tool				x	x	x	DIGAFLEX	6 & 7
Evaluation	Diagram	Firing rate analysis	Matlab	x				x	x	DIGAFLEX	6 & 7

* Project SmallCAN is to develop a low-cost and low-energy consumption system, in accordance to the works of [Diekhake, P., Liu, J., and Schnieder, E., 2011], [Liu et al., 2013] and [Diekhake and Schnieder, 2015]. This project is funded by European Regional Development Fund.
** Project EnEff:Stadt is to develop a integration methodology, including simulation, management and observation, for building, equipment and local infrastructure network [Kurczveil et al., 2014], funded by Federal Ministry for Economic Affairs.
*** Project DIGAFLEX is to construct a fieldbus-based intelligent control prototype, as a demonstration in building automation [Liu et al., 2013], funded by Federal Ministry for Economic Affairs and Energy and Energy.

1. Formulating the validation target by specifying the model concept. the attribute hierarchy method is selected and integrated with the OSI Model as the description mean. The model concept is structured with key properties, characteristics and quantities of each subsystem related to this work, in order to formalize the formulated concept under the context of system property categorization, as is mentioned in Section 3.1.
2. The quantitative analysis of timed parameter concerning concurrent-sending occurrences and corresponding access mechanisms are discussed in Section 4.1.

The data sorted from the log file have the same function relation structure of the real system, certain stochastic distribution of the message-sending frequency have been determined by means of goodness of fit, as shown in Section 4.6.

3. The Petri net communication model is constructed under the Petri net modeling platform named π-*Tool* [Quiroga, L. M., Becker, U., and Schnieder, E., 2014]. Then the model is parameterized with the timed parameters, based on the quantitative analysis of the concurrent-sending scenario with triggered access mechanism, as is discussed in Chapter 5. Monte-Carlo simulation is applied in generating the transition firing rates for further evaluation.

4. Extension and simulation of the Petri net communication model hierarchically regarding large scale number of SmallCAN fieldbus, further analysis based on the simulation results of the extended model is discussed under different traffic density in Chapter 6.

5. Function interrelations have been extracted and sorted from the fieldbus channel by analyzing the simulation results. This is discussed in Chapter 7.

3.4 Chapter Conclusion

The requirement specified in SmallCAN is formulated and formalized in this chapter. First, attribute hierarchy method is applied to structuring APL, DLL and the bus channel subsystems. The key properties, characteristics and quantities are expressed in order to formulate a formal specification of SmallCAN system. These specifications are discussed regarding the APL, DLL and bus channel subsystem of SmallCAN respectively in Section 3.1.1, Section 3.1.2 and Section 3.1.3. Third, the synchronization of subsystems mentioned above are discussed in Section 3.1.4, providing an overview of assignments concerning busload validation.

The categorization of these focused system properties are further specified regarding the validation procedure. As a result, the interlinking between a communication system and its means of formal descriptions is specified by the mapping between UMLCD and DSPN.

Finally, the BMW principle and related projects are selected to structure the validation procedure focusing on the system complexity.

Figure 3.6 profiles the specified tasks of this validation procedure by Petri net. It is worthwhile mentioning that specification plays a fundamental role. The SmallCAN model concept is generated from the specification. Moreover, the validated results retrieve and accomplish the SmallCAN specification.

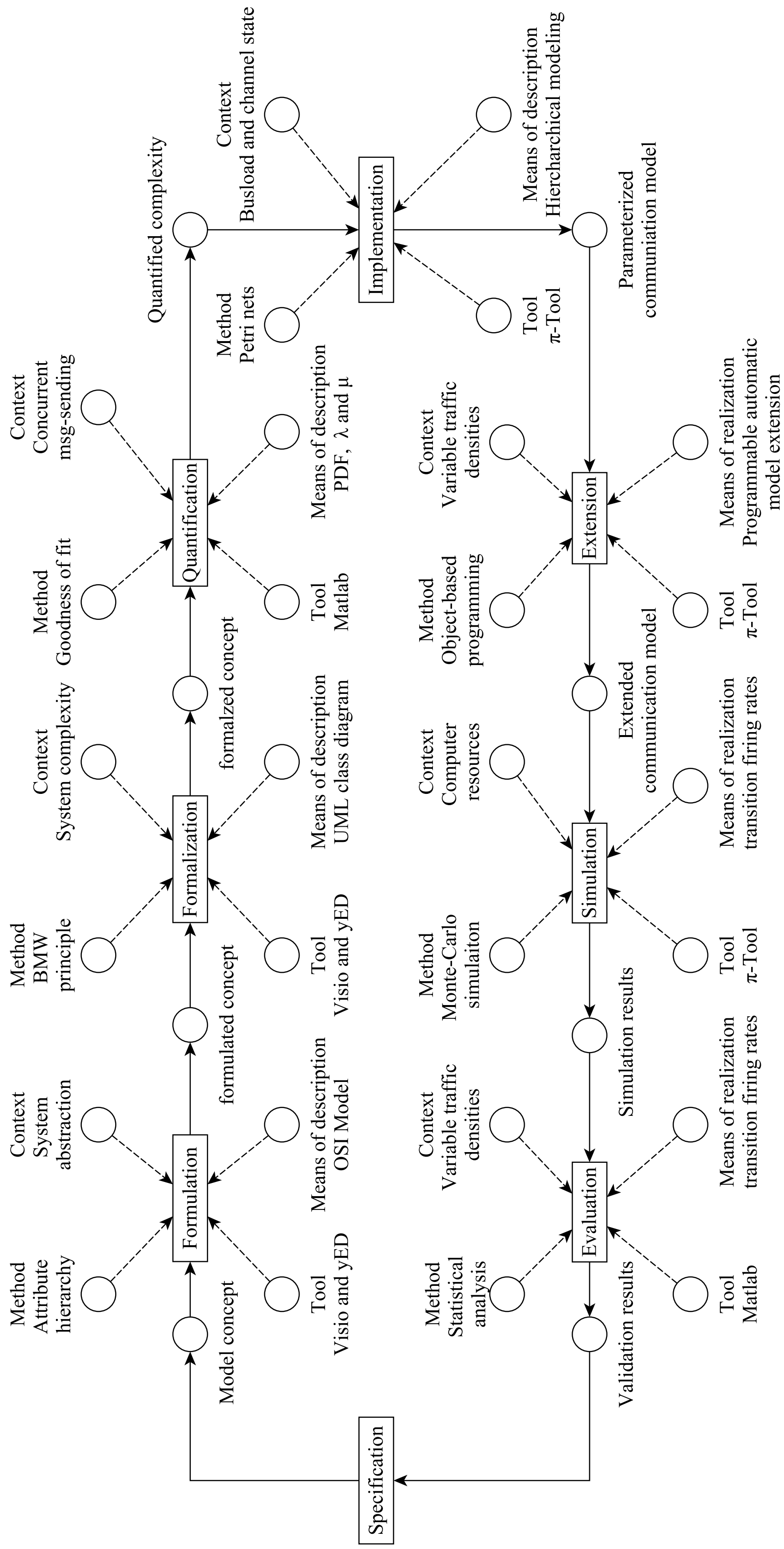


Figure 3.6: **Validation procedure by Petri net, based on [Schnieder, 2013]**

This validation approach is addressed to the platform of SmallCAN fieldbus system, which is designed and provided for fieldbus-based building automation by the Institute of Traffic Safety and Automation Engineering, Technische Universität Braunschweig. The protocol is inherited from the CAN bus-based protocol [Schrom, 2003]. The innovation of this work is to profile and quantify the system complexity as well as the corresponding number of function interrelations by formal methods, providing important criteria and findings for further system development and networking.

The steps of this modeling process are to specify the focused problem with attribute hierarchy, in order to have a comprehensive understanding of the key parameters related to fieldbus throughput validation. And the next step is to focus on these parameters as the system complexity resulted from the first step and describe it by quantification analysis. Within the sufficient context, the Petri net model can be generated and parameterized for further analysis.

Chapter 4

Quantitative Analysis regarding SmallCAN Busload

System complexity, represented by accumulating bus nodes and their functional interactions, causes the message burst on the fieldbus system channel. It requires reasonable focus on quantifying the concurrent sending occurrence.

The focus of this chapter is on quantifying the complexity concept with the attribute hierarchy theory on the system level, then the quantified analysis of the concurrent-sending occurrence, abstracted from the log data in real fieldbus-based building automation system. In addition, the timed parameters regarding message access mechanisms, such as CSMA/CA and bitwise priority comparison, have also been analyzed and determined. Furthermore, stochastic sending characteristics are quantified by the goodness of fit method.

4.1 Quantitative Overview of Busload

In serial communication, it is crucial to determine the minimum transmitting time interval. Regarding each specific fieldbus protocol, the prerequisite is to analyze the frame length as well as its variable structure. Table 4.1 shows a quantification review of difference fieldbus protocols. Requirement concerning timing in building automation is lower than the one in automobile. Therefore, $9.6kBit/s$ bit rate and Standardized Telephone cable ($2 \times 2 \times 0.06mm$) is applied in SmallCAN [Schrom, 2003]. Differential signaling is applied in Ethernet (Category 5 cable), Profibus and CAN.

During concurrent sending, CSMA/CA mechanism is adopted. The bus channel is accessed with the messages with higher priorities. On the aspect of accessed message, current information is not wasted. Therefore, it performs as a non-destructive bitwise arbitration role. On the contrary, CSMA/CD aborts all the concurrent messages back to their sources. Consequently, bus channel is not successfully allocated. To

	Bit Rate					Concurrency Solution						Error Checking				Bus Type			
	10 MBit/s to 10 GBits/s	10 kBit/s to 12 MBits/s	10 kBit/s to 1.0 MBits/s	1 KBit/s to 20 kBits/s	9.6 kBit/s	CSMA/CA	CSMA/CD	Master/Slave	Token Passing	Non-destructive Arbitary	Destructive Arbitary	CRC 32 Bit	CRC 16 Bit	CRC 8 Bit	Parity	Single Ended	Differential	Half Duplex	Full Duplex
Ethernet	X						X				X	X					X	X	X
Profibus		X						X	X	X		X	X				X	X	X
CAN			X			X	X			X	X		X	X			X	X	
LIN				X				X		X					X	X		X	
SmallCAN					X	X				X				X		X		X	

Table 4.1: Quantification overview of busload related characteristics and quantities among different fieldbus protocols

conclude, non-destructive arbitration, such as CSMA/CA and token passing, reserves the winning message in an effective way [Flammini et al., 2002].

By error checking part, LIN bus uses one parity bit for error checking [LIN, 2003]. CRC with 32 bit 16 bit and 8 bit error checking solutions has been applied to the rest fieldbus system mentioned in Figure 4.1. CAN adopts alternatively CRC 16 bit and 8 bit. CRC 8 bit is defined in the SmallCAN frame structure.

Serial communication systems, such as fieldbus SmallCAN, provide a single channel line to transmit and receive messages, in a non-simultaneously way. This is defined as half-duplex [Dias Pereira, 2004]. Ethernet and Profibus have a full duplex connecting topology. In this way, each node can simultaneously communicate with each other.

On the one hand, the message access mechanism is the resolution for balancing the bus channel traffic. On the other hand, it can be considered as another causal relation of generating extra message delay on the bus channel. Therefor,e the busload scenario with these access mechanisms needs first to be described and analyzed, from which the selected key-timed parameters can thus be determined. Thereafter the Petri net model of the worst-case scenario can be further extended and parameterized.

According to the validation method mentioned in Chapter 3, the first step emphasizes on describing the concurrent message-sending scenario with selected access mechanisms involved, as shown in Section 4.4. The timed parameters of which are abstracted regarding the fieldbus protocol defined in the DLL (Data Link Layer). So this step is focusing on the quantification and formulation of the concurrent characteristics of the the busload scenario. The results of this section can be further parameterized in the busload scenario model, as can be seen in Chapter 5.

The second approach of quantified analysis is to analyze the message-sending occurrences from diverse functional types of fieldbus nodes in Section 4.5. It is supported by the log data of message-sending occurrence recorded on the bus channel in real fieldbus-based system. The task here is to determine each stochastic distribution type as well as its PDF (Probability Density Function) of the focused message-sending node. The results of these PDFs quantifies the dynamic characteristics of the message-sending occurrence, This part is discussed in Section 4.6.

4.2 Requirements related to SmallCAN Busload

Figure 4.1 shows the relation of SmallCAN system and its busload by attribute hierarchy. The interactions between SmallCAN and its system environment generates the weather-changing event-based messages. These messages can be characterized either in a stochastic or a deterministic way. In SmallCAN system, messages caused by weather changing are mainly stochastic events. This part is analyzed and discussed in Section 4.5.

Functions sequences also generate message sequence on the bus channel. It take a deterministic time interval of executing a fixed path of the function architecture predefined inside the Application Layer.

As a result, the busload is accumulated of the quantities of stochastic and deterministic message sending as well as the time interval of message-sequence.

The busload performs as two behavioral properties, which are concurrency and emergent bus behavior.

The concurrency in serial communication is inevitable, so it can be solved with Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) and bitwise arbitration. They are in charge of scheduling and balancing the busload caused by concurrency, which are introduced with scenarios in Section 4.4.

The emergent bus behavior can be characterized into overload and bust. Overload is the worst-case scenario for throughput of the fieldbus system. Fieldbus overload scenario can be detected by channel utilization. As for SmallCAN, the theoretical overload limit is calculated in Section 4.3.

Burst is another worst-case scenario related to executing the function sequence. The upper limit of this value is defined as the maximum allowed time interval observed

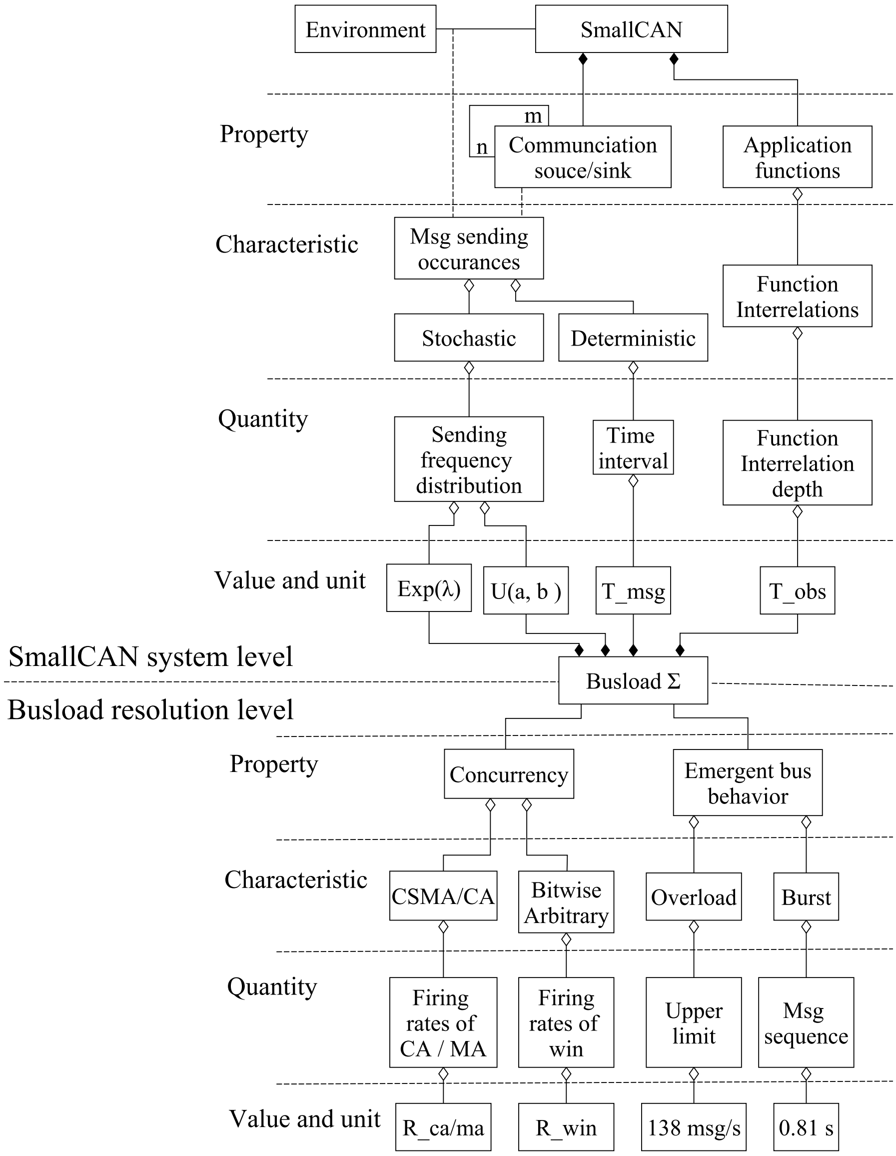


Figure 4.1: **Requirement specification of quantifying SmallCAN and its busload**

in real system, which can be further set as the SmallCAN requirement. This part is further discussed in Section 7.2.

4.3 Quantification of SmallCAN Frame Structure

The prerequisite of analyzing the SmallCAN busload is to investigate the frame structure in the SmallCAN protocol.

A SmallCAN frame is mainly composed of 16 bits of address field, 16 bits data field and 8 bits CRC (Cyclic Redundancy Check). Furthermore, synchronizing bits, *i.e.* Start-Stop bits, are inserted into each bytes of the SmallCAN frame, defined in [Schrom, 2003]. The calculation results between these the SmallCAN frame and a CAN frame is presented in Table 4.2.

A comparison of frame structures between SmallCAN and CAN fieldbus system is shown in Figure 4.2. Compared with CAN frame, the data field of a SmallCAN frame is optimized as a fixed length [Schrom, 2003] for building automation. In addition, the arbitration field of a SmallCAN frame is longer than CAN. As a result, a coverage of more SmallCAN messages IDs is available in building automation systems. CRC is in charge of detecting the error during the physical transmission. Unlike the relatively higher standards defined in the field of automotive, the length of CRC in the SmallCAN frame is 8 bit, compared with 16 bit length of CRC in CAN frame.

The advantage is that message errors can be identified effectively and separately by misusing or missing the start or stop bits in each part. According to Table 4.2, the fixed length of the message in the SmallCAN specification is summed up to 69 bit.

Table 4.2: Frame length calculation of a SmallCAN and a CAN frame

Section	SmallCAN / bit	CAN / bit
Start-of-Frame	1	1
Arbitration field	$16 + 2 \times UART^{*}$	$11 + RTR^{**}$
Control field	0	$4 + IDE + R^{**}$
Data field	16 (fixed) + $2 \times UART$	$0 - 64$
CRC	8	16
End-of-Frame	$6 + 12$ *Silence*	$7 + 3\ IFS^{***}$
Total	69	$44 - 108$

Note:
* Universal Asynchronous Receiver/Transmitter, Start-Stop: $1 + 1.25$ bits.
** RTR (Remote Transmission Request), IDE (Identifier Extension bit), Reserved bit.
*** IFS (Intermission Frame Space): 3 bits [Bosch, 1991].

The goal is to avoid two or more messages colliding with each other during channel occupation. Therefore, the minimum time interval of one channel occupation message is theoretically calculated in Equation 4.1.

$$T_D = \frac{FrameLength}{BitRate}. \tag{4.1}$$

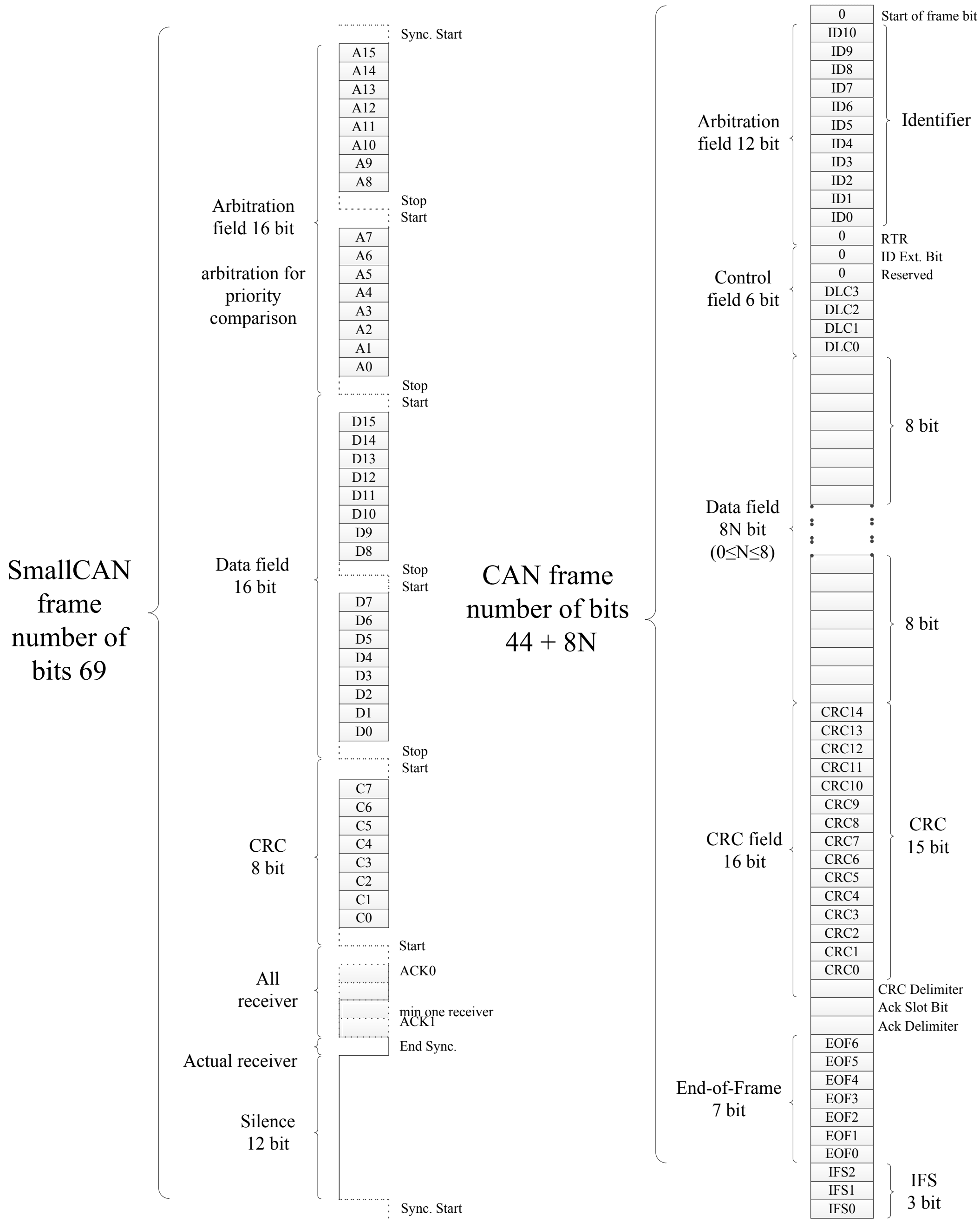


Figure 4.2: Frame comparison between SmallCAN [Schrom, 2003] and CAN

According to SmallCAN specifications, the bit rate is fixed. So the minimum time interval between two occupying messages is calculated by Equation 4.2.

$$T_D = \frac{69\ bit}{9600\ bps} = 7.1875ms. \tag{4.2}$$

4.4 Message Access Mechanisms Integrated in SmallCAN

The concurrency of low-quality and high-quality channels of MAC poses problems of fairness and efficiency in networks [Ha, H., Wang, Z. and Lee, J, 2014].

CSMA is a mechanism of message access involving probabilistic behavior. It performs in which each fieldbus node monitors the channel state before sending the message to occupy the channel. The application fields of different CSMA categorization is listed as follows:

- CSMA/CD (Collision Detection) protocols for collision detection and involving a retransmission mechanism. It is used in Ethernet bus and hub networks.
- CSMA/CA protocol for avoiding collisions by a waiting time after detecting the channel free state. It is also applied in wireless LAN.

The priority comparison mechanism is integrated with CSMA/CA (Courier Sense Multiple Access / Collision Avoidance) mechanism in SmallCAN [Schrom, 2003]. The goal of this Section is to describe these message access mechanisms and further integrated into the Petri net communication model.

Bitwise arbitration mechanism is based on the priority difference to decide which frame is permitted to transmit. The goal is to deal with concurrent message-sending. The bitwise arbitration mechanism is described in Figure 4.3. Three frames with different priorities simultaneously request to occupy the bus channel.

The priority is assigned to each frame with a sequence of bits in the frame identification part. The bitwise comparison occurs only in this part of frame. Regarding the frame structure in SmallCAN, the priority part occupies two bytes, the frame structure of which is further showed in Figure 4.2. In this case study, the priorities of three frames are defined as :$P_A > P_B > P_C$.

As can be seen in Figure 4.3, this priority structure is represented and executed by the recessive bit (logic 0) and dominant bit (logic 1). During arbitration, the dominant state overwrites the repressive state. As is shown in Figure 4.3, three frames simultaneously start to transmit. Therefore, they begin to enter the bitwise arbitration at time stamp t_0. At t_1 Frame B toggles its bit to logic 0 while Frame B and C still remaining logic 1. At this moment, Frame B is lost to Frame A and Frame C. As a result, the further content of Frame B will not be appeared on the channel and sent back to its source. The similar scenario occurs at t_2, Frame C is lost to Frame A because of recessive bit.

It is necessary to be mentioned that arbitration compares the bit level of every transmitting frame with the bit level observed on the channel [Bosch, 1991]. Recessive bits are overwritten only at the moment that the others remain dominant bits. Therefore, this will not destroy the frame information of the winner to transmit on the channel. The mechanism guarantees only one message remaining active. In order to proceed quantified analysis, the message access mechanisms are interpreted by the time digram of concurrent message-sending scenarios described in Figure 4.4.

Figure 4.4 shows that four groups of message sources request to send spontaneously on the fieldbus channel. Their priorities are decreasing in accordance with the descending order similarly defined in the ID section of the message captured from the fieldbus system in reality. The definitions and descriptions of time parameters together with their constraints are listed in Table 4.3, concerning message access mechanism inside this collision scenario. There are totally four messages requested to occupy the bus channel. Six message transmission scenarios, two main concurrent

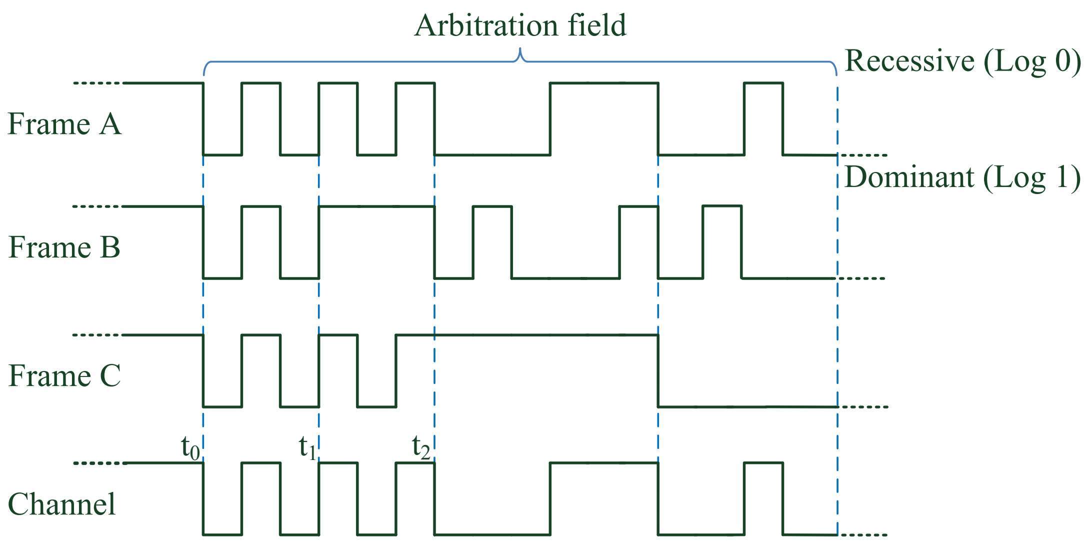


Figure 4.3: Bitwise arbitration mechanism in SmallCAN

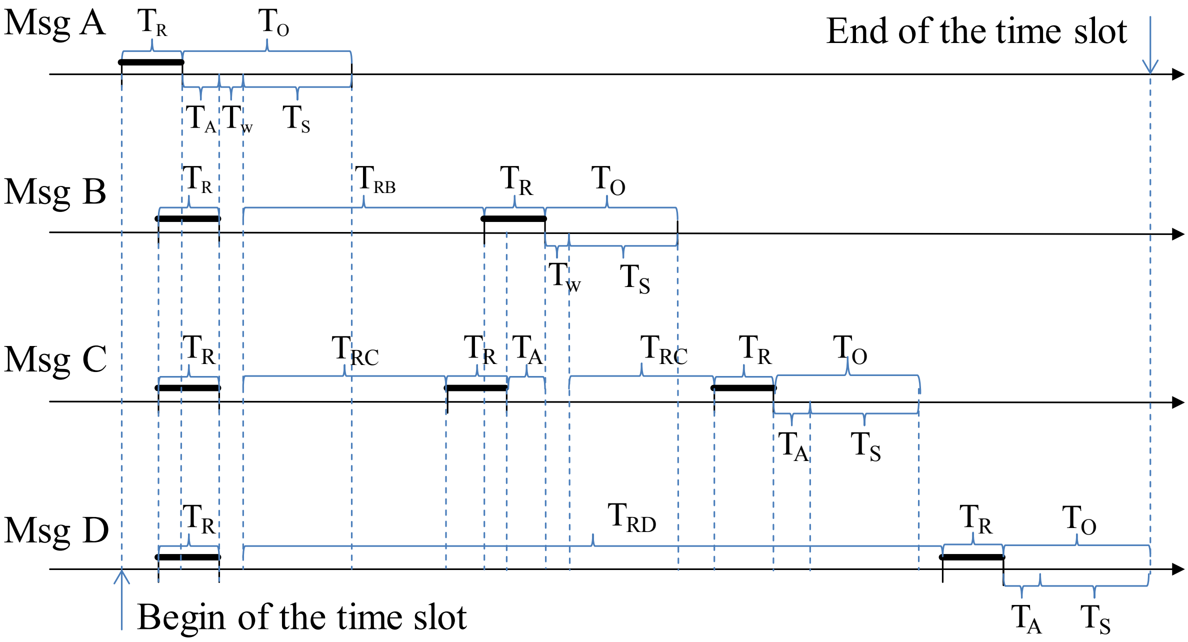


Figure 4.4: **Time diagram of collision scenario**

Table 4.3: **Timed definitions of the message collision scenario**

Name	Description	Condition
T_R	Request to occupy the bus channel	fixed
T_O	Operation time interval of message access	variable
T_A*	Single access to the bus channel without conflict	slightly variable
T_W**	Time interval of the "winning" due to priority comparison mechanism	slightly variable
T_S	Confirmation to occupy the bus	fixed
$T_{R_{B,C,D}}$***	Time interval for retransmission after "losing" due to priority comparison mechanism	variable

Note:
* These timed parameters are subject to the CSMA/CA access mechanism defined in Norms such as IEEE 802.11 in wireless network [IEEE802.111, 2012].
** These timed parameters are generated from the CAN norms, such as [Hartwich, 2012].
*** The retransmission timings obey the rules of the CSMA/CD defined in Ethernet norms, such as IEEE 802.3 [IEEE802.3, 2016].

message transmission, four retransmissions together with two single message transmissions are chosen due to various combinations of these three access mechanisms defined inside fieldbus requirements concerning collision scenario.

The case study is shown in Figure 4.4, normally message to be sent on the bus should take T_A for single access waiting process on condition that the bus is currently free. In addition, after any message finished waiting T_A in single access and has been sent to the next step for bus channel occupation, which are directly sensed by the rest concurrent-sending messages. Therefore, T_A can be skipped, instead of waiting in single access, they are immediately sent to the bus for further procedure due to the multiple access described in message access mechanism. Priority of the message is defined bit by bit in the address part, *i.e.* the bigger address value means the higher priority for sending in privilege. In this case the order of $Pri_A > Pri_B > Pri_C > Pri_D$ is set. The time T_W occurs shortly because of the bit by bit priority comparison in the message address part to decide which message would win and successfully occupy the bus channel and the rest to go the retransmission phase. Due to the limit of context, the retransmission phase is discussed in Chapter 5.

The time duration after the completion of request to occupy the bus channel until the final occupation phase of one message is defined as operation time, it could involve different circumstances during the entire message transmission process, Therefore, the time T_O can be generally defined by Equation 4.3 based on three different situations.

$$T_O = \begin{cases} T_A + T_W + T_S & collision \ \& \ arrives \ early \\ T_W + T_S & collision \ \& \ arrives \ late \\ T_A + T_S & bus \ free \end{cases} \tag{4.3}$$

The first row of Equation 4.3 can be interpreted by message A in the first round all concurrent-sending messages mentioned in Figure 4.4. Request to send time T_R occurs firstly to message A and after a short time, T_R simultaneously occurs to the rest of three messages. Since the moment that message A finished, T_A as single access waiting time is sensed by the rest of these three concurrent-sending messages. They are also brought to the next process due to multiple access without waiting for access. Then message A wins out with time T_W and successfully occupies to the bus channel with time T_S. Because of its defined dominant priority, the rest three later requesting messages are sent back to their message source, *i.e.* the bus nodes generating T_R as waiting time for their retransmission. In this conclusion, message A arrives early and wins out for occupying the bus channel.

A good case study for comprehending the second row of Equation 4.3 is message C. After its lost in the first round of priority comparison due to its relatively lower priority defined, after the first retransmission time T_{RC} is consumed, message C requests to occupy the bus channel earlier than the same process of retransmitted message B. The moment that message C finished its waiting time T_A as single access mechanism is sensed by all the other concurrent-sending messages. In this case the second round transmitting message B, it is immediately sent to the bus channel. Due to the triggered multiple access mechanism defined in CSMA/CA, message B is directly sent on the bus for further priority comparison without spending waiting time T_A. Finally, message B will win out with relative higher priority than message C, it should be mentioned that T_O only contains the waiting time on aspect of successful transmitting message. Therefore, T_A is herein neglected. And collision doesn't occur to the rest of two sending scenarios of message C and D. Only T_A and T_S are hereby involved for calculating operation time.

Slight variation described inside T_A in Table 4.3 means that on the collision state, it might occur that multiple access can trigger even though the waiting process of a for-single-access message already starts to request for the channel access. Then these two or more messages can be send to the channel for further authorization without waiting. This situation alleviates the waiting process of T_A with the respect of the message experiencing only the part of Single Access. Therefore, one uniform distribution between the range of 0 and T_A is defined in each independent transmitting the message source. By contrast, the slight variation of T_W is also a result of the uncertainties occurred during the collision scenario. The message can immediately win out during priority comparison, when its first address bit differs with others' due to bit by bit comparison. Therefore, another uniform distribution is introduced, the scale of which is depending on the length of the frame, data rate and traffic density.

It is concluded that the relationships of time quantities in two collision scenarios and two idle scenarios are described and analyzed. It also needs to be mentioned that the retransmitting time duration of one message, which eventually occupies the bus channel and begins to transmit in principle, is summarized in Equation 4.4, where N_R is the number of continuous retransmission times.

$$T_{busy} = \begin{cases} (N_R + 1)\ (T_R + T_O) & N_R \in \mathbb{Z}_{\geq 0},\ Msg\ win \vee idle \\ (N_R + 1)\ (T_R + T_W + T_{R_{B,C,D}}) & N_R \in \mathbb{Z}_{\geq 0},\ Msg\ lose \end{cases} \tag{4.4}$$

4.5 Message-Sending Occurrence in Real SmallCAN System

Sending frequencies of all field bus nodes play a significant role for busload modeling. Every message generating event has to withstand all related message access mechanism in accordance to specified fieldbus system requirement, all these transmitting events act firstly as input of the busload traffic, additionally they also interact with each other when concurrency scenario occurs under relatively high traffic. So it is necessary to discuss the message-sending occurrence based on the message behavior of fieldbus system in reality. The traffic of independent message transmitting event is discussed in the following section. Message sequence and its relations will be discussed in next chapter.

On the aspect of busload in closed communication transmission systems, stochastic and deterministic message-sending occurrences compose of the throughput in the fieldbus system. Especially in the application context of fieldbus-based building automation, busload tended to consisting of more environmental-based message-sending. Therefore, the categorization of message types needs to be carried out. Message-sending types of most frequent transmission events regarding this fieldbus system have been classified into three groups.

1. Passive message sources depending upon the system input from the environment, such as temperature sensor, moving detecting sensor and so on.
2. Active message sources defined in fieldbus requirement to control or monitor the system QoS, for example the cyclic messages from the server to check the fieldbus nodes' state.
3. Synthesized message sequence as important piece for generating busload as message sequence in accordance to the function relationship defined inside the Application Layer, a good case study is FSF concept mentioned in Section 3.1.1, which will be discussed and validated in Chapter 7.

The research on the characteristic of event-based message-sending plays an important role in accumulating the busload. Therefore, it is necessary to quantify it as a key part of busload validate procedure. A case study is carried out, including of the messages controlling temperatures, lightening and anemometers compose the most traffic densities on a daily basis, although these passive message sources generated from the environment are slightly different with each other. Therefore, it is of absolute necessity to find out the message-sending occurrence in the primary types of message sources for busload validation.

The message-sending characteristic of temperature message sources can be partially mapped by the temperature message-sending occurrence shown in Figure 4.5. Each message generating event ascribed by environmental temperature change has been transmitted on the bus channel and recorded into fieldbus server as log file. The data inside log file can be further analyzed, sorted and plotted as the whole sending occurrence of temperature message sources on a daily basis. In this work, seven temperature message IDs are selected among twenty temperature sensors implemented in the fieldbus-based smart office, in order to represent impact of the environmental triggered traffic density on the fieldbus channel. These seven selected messages' IDs indicate seven message sources from the temperature bus couplers installed inside the office wall implemented with SmallCAN fieldbus system. Therefore, any slight temperature change inside the office room, the value of which is larger than an offset

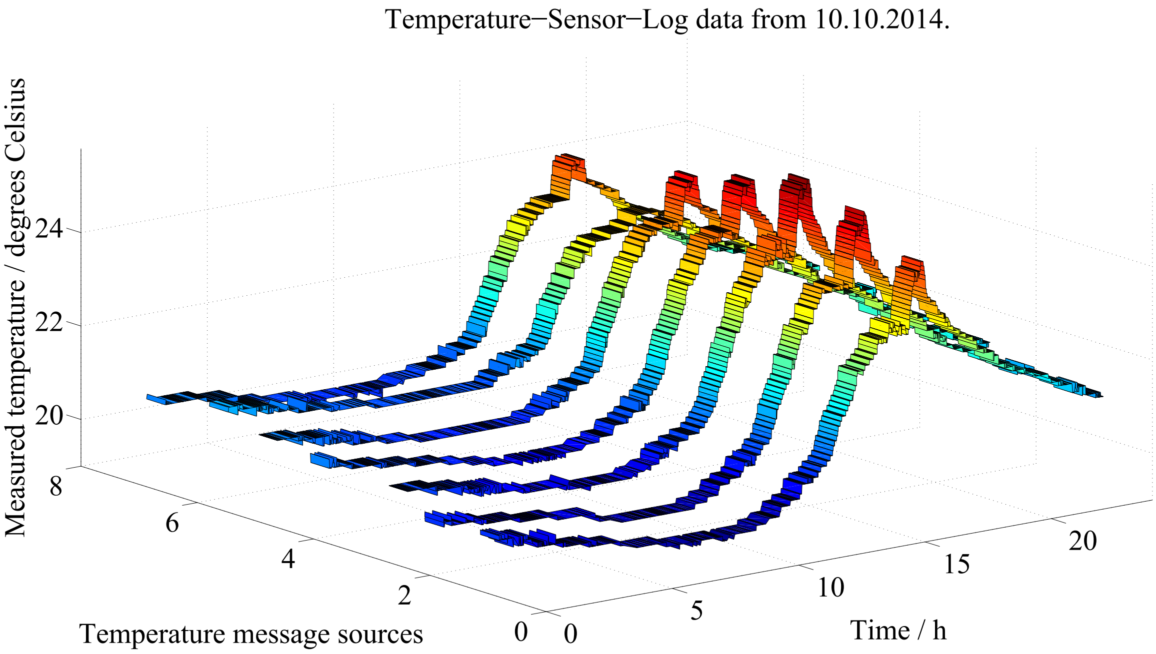


Figure 4.5: **Temperature messages plotted from log data**

defined in the coupler's code for message traffic control, has further been chronically recorded, rearranged and shown on this figure over the whole day. The offset concept is implemented in the coupler's software part. Each temperature coupler is designed to have lower traffic impact on the bus channel. Therefore, the adequate change in the temperature value is detected, calculated and omitted to send on the bus. In addition, the office temperature state in this case is influenced with the combination among climatic change, air conditioning system as well as human involved activity.

Based on the technical context mentioned above, seven parallel curves are plotted under three coordinates. They have been specified as 24 hours over one day, temperature message sources and degrees Celsius. The focus of posting the degrees of centigrade is tantamount to indicate the fact that more frequent changes in the environmental temperature result in more frequent messages generated on the bus channel. The time spans of most frequent message-sending events as well as temperature value peak mostly occur from 10 o'clock to 20 o'clock. Because the message traffic on the bus is mainly generated by human involved activities together with environmental changes at that time scale. Even all these seven curves are generally subject to the same temperature changing tendency. Some undistinguished differences exist because of the temperature sensors' location together with different influence on these combined parameters mentioned above. The sixth temperature message-sending curve on Figure 4.5 refer to the temperature sensors mounted relatively on the bottom of the wall, whereas the last temperature sending curve with recovered ascend temperature peak is mounted near at very bottom and near the pipeline of the air conditioner.

Temperature characteristic has been mapped by this temperature-sending occurrence in 3-dimensions. Around 18 °C as one extreme temperature value occurs between 3 o'clock and 7 o'clock colored with black and dark blue while 26 °C as another extreme temperature value occurs at noon.

Nevertheless it is worthwhile to point out that the time stamp of the temperature peak value has overlapping with the time stamp of the peak value of message-sending frequency, this can be explained as follows:

- Value-calculation methods are different between these two time spans. In other words, the message-sending moment of the highest temperature value differs from the message-sending moment of the highest frequent value.
- The temperature value is directly detected by the temperature sensor itself. For example, if the temperature peak value in the noon remains within a certain range, this would remain relatively high while the correspooning message-sending frequency in this case is not the highest or even relatively low.
- The message-sending frequency is calculated by sorting the absolute time stamps between every two message-sending events in the log file. For example, only if the environmental temperature changes quickly, can the corresponding message-sending frequency remain relatively high.
- The fitting process of the message-sending frequency is further discussed in Section 4.6, in order to find the rules of the focused message-sending frequency in either a deterministic or stochastic manner of this case study.

4.6 Fitting Stochastic Distribution Types of Real Message-Sending

Sending Event Nr.	Abs. Time	Msg ID	Msg Value
1	339	32767	5513
2	419	517	585
3	863	8303	10205
4	884	8302	8199
5	884	8301	8468
6	910	8300	4352
7	1330	32767	5514
8	1710	692	256
9	2349	32767	5515
10	2369	8203	10205
11	2370	8202	8199
12	2389	8201	8468
13	2390	8200	4352
14	3341	32767	5516
15	3419	517	586
16	3719	692	261
17	4352	32767	5517
18	4762	8213	49083
19	5332	1799	36160
20	5349	1798	36183
21	5351	32767	5518
22	5431	8217	33185
23	6052	1515	21272
24	6342	32767	5519
25	7359	32767	5520
26	7439	8217	33408
27	8359	32767	5521
28	9351	32767	5522
29	9399	712	33087
....	....	....	
139190	62581432	6995	512
139191	62581445	2812	32938
139192	62581451	1812	256
139193	62581481	15499	512
139194	62581491	15500	512
139195	62581502	2888	32946
139196	62581511	2811	32930
139197	62581513	1813	256
139198	62581521	1811	256
139199	62581542	4077	32897
139200	62581550	1072	256
139201	62581592	4013	512
139202	62581593	4012	512
139203	62581600	4011	512
139204	62581611	2010	256
139205	62581661	32767	2565
139206	62581681	4010	512
139207	62581692	4009	512
139208	62581694	2009	256
139209	62581731	4028	512
139210	62581751	4027	512
139211	62581753	2028	256
139212	62581780	4017	512
139213	62581792	4016	512
139214	62581800	2017	256
139215	62581831	4006	512
139216	62581841	4005	512
....	....	....	

Figure 4.6: **Log data sample**

The log file data observed from SmallCAN bus server consists of three columns representing three characteristics of all recorded message events occurs on as the fieldbus traffic.

As shown in Figure 4.6, log data are composed with absolute time stamps, message IDs and message information part. Each row of the log file indicates one message-sending event captured on the bus channel. Based on these mentioned characteristics, it is available to analyze the sending characteristics of selected message IDs and then convert them into the busload model for message-sending parameterization. Therefore, the data analysis work focused on log file data has been carried out for plotting result shown in Figure 4.5 and Figure 4.7.

The focus in this subsection attempts to find the timed rules of the selected message-sending characteristics by sorting the log file data. Therefore, the first two columns are chosen due to their timed interpretation and message types while the third column of the log file matrix is intuitively neglected due to its functional representation based on message IDs.

The data-sorting method for quantifying the same function interrelations is introduced by following steps:

1. traverse the log file, refine the matrix with the selected message ID. Iterating the second column of the matrix with the selected message ID, remain the relevant rows and delete the rest rows. Then the characteristic of absolute time stamp with its message ID has been refined.

2. calculate the relative time span between every two messages in the refined matrix, *i.e.* the time interval of every two rows indicating every two messages-sending events with same selected ID results is calculated by the abstraction of every two absolute time stamps recorded in first column in log file. The message-sending frequency from the selected message source can be further concluded based on the relative time spans. It is crucial as a part of the system input from the environment for further validation process.

3. sort the rest selected message IDs with the same method based on first two steps mentioned above. In order to approach the worst-case scenario of busload modeling, one message source in this case represents the integration of all selected message IDs subject to the identical functional characteristic. So it is approachable to describe the message-sending characteristics with the sufficient key nodes included to quantify the system complexity.

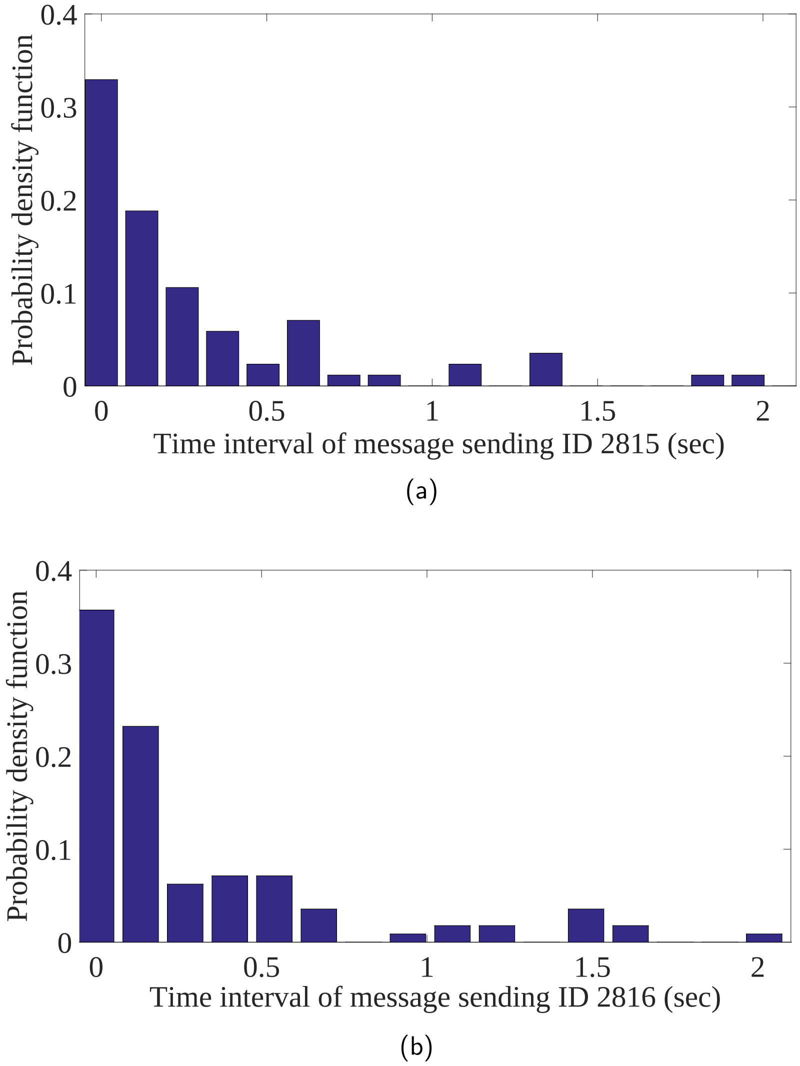


Figure 4.7: **PDF Histogram of sorted stochastic message-sending IDs**

4. plot and distribute these timed results calculated in first three steps into calibrated time bars, then the frequency of each time bar is acquirable by this data plotting method, in this histogram the stochastic distribution type of selected message source has been further fixed.

5. plot all relevant types of stochastic distributions together with the histogram mentioned in the fourth step in order to fit the certain distribution type with the similarity indicators provided.

Based on the first three data sorting steps mentioned above, the message-sending behavior is gathered and sorted into histogram to reflect the stochastic characteristics of selected message sources in the refined log data, shown in Figure 4.7.

Message IDs 2815 and 2816 are typical temperature message sources selected among these seven temperature message sources in Figure 4.5. Among the three coordinators, the z-coordinate represents the message information part. This is hereby neglected because of the focus on the message-sending frequency.

The x y coordinate axises are introduced as follows:

- The X-axis represents the time interval between every two message-sending occurrences, the reciprocal of which is the message-sending frequency.
- The Y-axis describes the PDF of these message-sending occurrences.

Therefore, the given PDF integral of variable the message-sending probability density with respect to message time interval equals to 1. As can be seen in both histograms, most message-sending occurrences are within the time interval of $100ms$. And most messages occur within $150ms$, probability density of message-sending is downward trend with regard to the time interval in both curves.

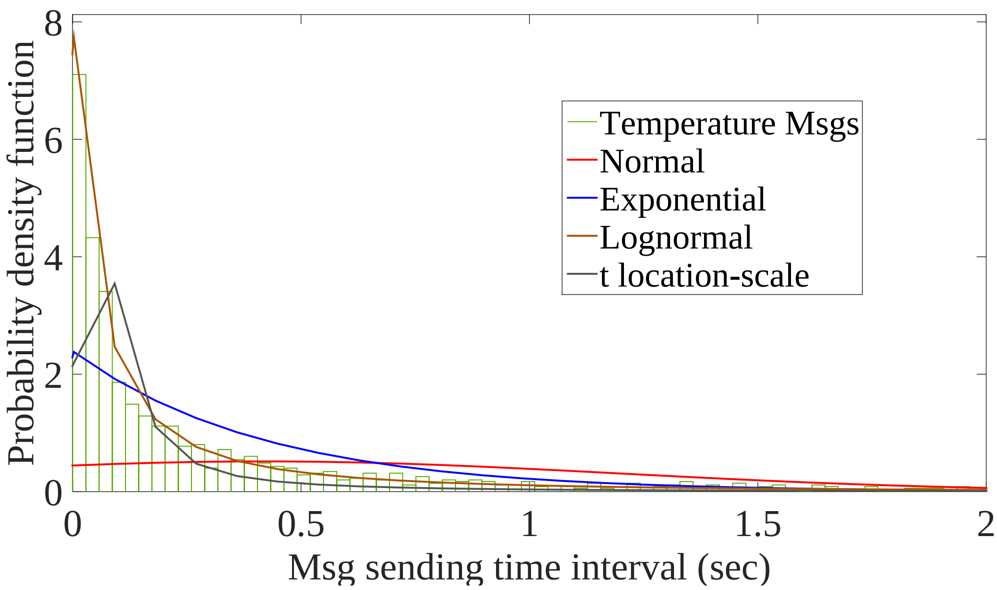


Figure 4.8: Fitting the stochastic distribution types of the real message-sending

Stochastic message-sending behavior can be seen in both message sources in Figure 4.7, although they slightly differ with each other. Therefore, seven temperature message sources are integrated into one synthesized histogram with more approximation to the message-sending behavior in average.

As is shown in Fig 4.8, Goodness of fit method is applied in fitting the focused curve with different stochastic distribution types. The similarity between each fitting approach and the focused curve is calculated by the likelihood value as the result of each fitting approach.

Table 4.4: Fitting results of selected distribution types

Color	Distribution type	Quantities	Likelihood
Green	Need to be fitted	Expectation and variance	
Red	Normal	$\mu = 0.42, \sigma = 0.77$	-1402.92
Blue	Exponential	$\lambda = 2.39$	-153.55
Brown	Lognormal	$\mu = 0.55, \sigma = 5.78$	175.17
Black	t location-scale	$\mu = 0.065, \sigma = 0.07$	-449.81

Therefore, fitting criteria of several appropriate distribution types with corresponding likelihood values are given and listed in Table 4.4. Finally, the highest likelihood value provided by lognormal distribution type is selected. Its quantities can be further applied in the mathematical analysis of concurrent sending. Moreover, expectation and variance of fitted lognormal distribution can be parameterized as the Petri net model input for the further validation steps.

4.7 Mathematical Analysis of Channel Concurrency

The mathematical deductions in this section are focusing on the timed analysis of the channel collision. Therefore, the CSMA/CA and bitwise arbitration are not considered.

As is mentioned in Section 4.3, the minimum time interval between two sending messages among all sending nodes in SmallCAN is greater than or equal to $7.1875ms$. Therefore, the channel concurrency in this case is expressed with Definition 4.7.1.

Definition 4.7.1. Channel concurrency is theoretically defined that the second or more transmitting messages are simultaneously sending or time-wise overlapping with the first transmitting message on the bus channel.

The probability of the channel concurrency is expressed in Equation 4.5,

$$P_{\_Collide} = \frac{n!}{m!\,(n-m)!} \int_0^{t\_min} f\,(t)\,dt \tag{4.5}$$

where

- n represents the total number of messages from the independent message sources successfully occupied the bus channel.
- m represents the current transmitting messages on the bus channel within the time interval $t\_{min} = 7.1875\ ms$.
- $\frac{n!}{m!\,(n-m)!}$ represents the possible combinations of further concurrent-sending within the time interval $t\_{min} = 7.1875\ ms$.

$f(\sigma)$ is defined with lognormal PDF in Equation 4.6.

$$f(\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}} \tag{4.6}$$

The prerequisite in Equation 4.5 indicates that the messages have successfully authorized the channel occupancy for further transmission. Therefore, the access mechanism implemented before bus occupation is herein neglected. With the fitted PDF of the message-sending frequency mentioned Section 4.6, it is available that the probability of more channel collision scenario can be calculated based on this mathematical model.

Two notions with regards to fixing the stochastic distribution type mentioned in Figure 4.7 are illustrated here:

1. $f(\sigma)$ is the probability density function of the message-sending frequency, which distribution type would be fixed according to the sorted message-sending frequency from the message source in fieldbus-based building automation system, as shown in Figure 4.7 and Figure 4.8. The frequency is sorted with a group of the same type nodes as message source rather with the fieldbus node-oriented sorting. Only by fitting the frequency distribution type from the real fieldbus system, can the Petri net model input be better approaching to the real sending behavior.

2. By combining the fitting quantities of lognormal distribution of the focused curve, the channel concurrency possibility can be calculated in Equation 4.7,

   where a simplified concurrent scenario with maximum networking messages is presented, with the given condition of $n = 1000,\ m = 1$.

   $1/\lambda$ represents the expectation of the sending time interval between every two messages. In this case, the average value $1/\lambda = 50\ msg/s$ is calculated from these two message-source-nodes. Therefore, on the condition that the expectation value of the sending time interval equals to $0.02s$, the probability of two message collision of theoretical value.

The result of Equation 4.7 can be further calculated with the fitted quantities shown in Table 4.4.

$$P_{\_Collide} = \frac{n!}{m!\,(n-m)!} \int_0^{t_{\_min}} f\,(t)\,dt \tag{4.7a}$$

$$= \frac{1000!}{1!\,(1000-1)!} \left[F(t_{\_min}) - F(0)\right] \tag{4.7b}$$

resulting in the CDF shown in Equation 4.8.

$$F(t) = \frac{1}{2} + \frac{1}{2} erf \left[\frac{lnt - \mu}{\sqrt{2}\sigma}\right] \tag{4.8}$$

In case of exponential distribution shown in Figure 4.7, the PDF is defined as follows:

$$f(\sigma) = \lambda e^{-\lambda\sigma} \tag{4.9}$$

and the probability of channel concurrency with $\lambda = 0.02$ can be calculated as follows:

$$\begin{aligned} P &= C_{1000}^{1} \int_0^{0.007185} \lambda e^{-\lambda\sigma} \\ &= C_{1000}^{1} \int_0^{0.007185} 0.02 e^{-0.02\sigma} d\sigma = 14\% \end{aligned} \tag{4.10}$$

4.8 Chapter Conclusion

First, with the help of structuring the requirements related to SmallCAN busload generation and resolution, the dynamic behavior between the SmallCAN system and its context has been characterized by the attribute hierarchy. Based on this, a quantitative analysis of the selected mechanisms in MAC, such as CSMA/CA and bitwise arbitration, has been carried out.

Second, for the quantitative analysis in real SmallCAN system, the temperature related message-sending behavior has also been profiled by plotting the relevant SmallCAN log data. Moreover, time interval of two typical message IDs with their relevant PDFs have been further abstracted and sorted.

By fitting the stochastic behavior of message-sending events, the goodness of fit method has been applied to fitting distribution types of the stochastic behavior of the message-sending in real SmallCAN system given by the most likelihood values, including fitting the quantities, such as expectations and variances.

Besides, channel concurrency has been theoretically defined and mathematically deducted based on the fitting results.

The results of the quantitative analysis can be seen as the validated timed-criteria improving SmallCAN protocol. They are crucial for the further validation procedures, such as the parameterization of the focused busload Petri net model.

Chapter 5

DSPN Modeling of the SmallCAN Communication

According to [Zimmermann and Hildebrandt, 2015], Petri nets as one of the formal classical modeling means are distinguished for analyzing the dynamic behavior of complex systems. The causal and temporary relations can be integrated and well described in the syntax and semantics of Petri nets. The compositionality of Petri nets means that modular sub-nets can be hierarchically combined to form a complex Petri net. The goal is to allow the construction of a large communication model by using a number of parallel sub Petri nets which are interactive and interrelated with each other in a well defined way [Schnieder et al., 2009].

As carried out in Chapter 4, the scenario assignments for modeling the communication system by DSPN is outlined in this Chapter.

The Petri net communication model is introduced with two kinds of sub-models: channel state sub-model and message-source-node sub-model. The explanations of Petri net elements and their interactions involved in constructing mapping between the net structure and the focused system procedure are hereby also specified.

With the help of computerized simulations and the adjustable transition rates, values of Petri net model, such as transition firing rates and place occupation rates, can be generated and further sorted for data analysis: Chapter 6 and Chapter 7 focus on this. Transition firing rates with regard to critical dependability of the communication model are mainly generated and analyzed, providing quantified measurement during the system validation and verification phase.

5.1 Overview and Assignments of Modeling SmallCAN Communication

With the aim of modeling a worst-case scenario of the message burst on SmallCAN fieldbus channel, Petri net modeling as one of the formal approaches is used to build the concurrent-sending scenario. The goal is to make the communication model mapping the collision scenario occurred on the real fieldbus channel, the time related parameters have been selected, prioritized and analyzed in Section 4.1. Here the Petri net model can be constructed and parameterized either with model input or its own modeling structural components. E.g. parameters related to the CSMA/CA mechanism as well as the bitwise priority comparison are integrated into the model structure. Moreover, the fitted stochastic distribution type of message-sending frequencies in real system mentioned in Section 4.5 are hereby mapped to the transition firing delays. This can be seen as the model inputs of generating the busload onto the channel. The whole communication model in this chapter is hierarchically built with the Petri net platform π-*Tool*.

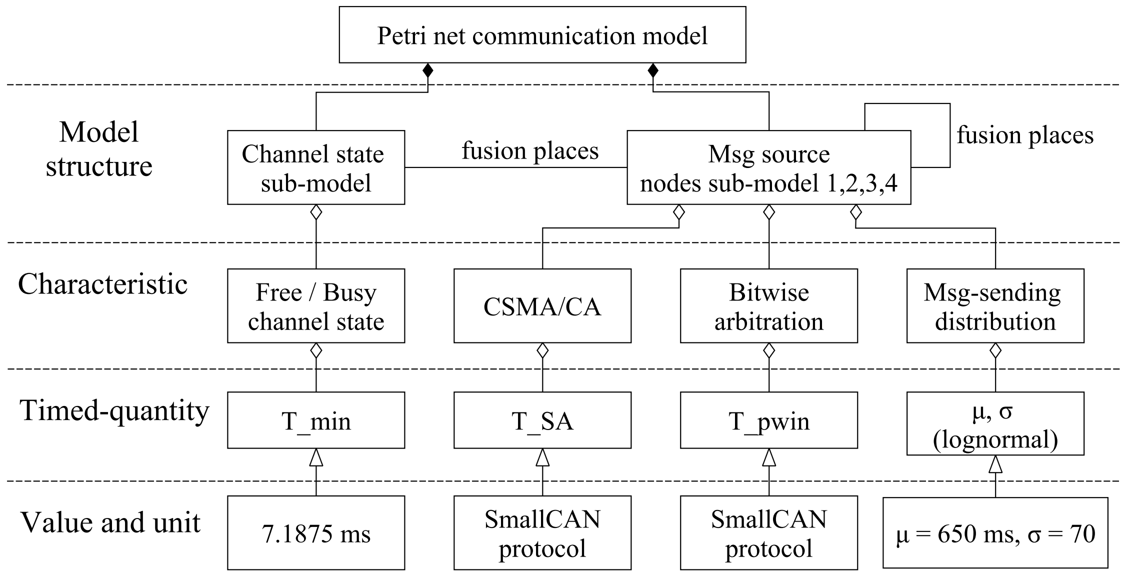


Figure 5.1: **Overview of modeling structure by attribute hierarchy**

As mentioned by Table 5.1 in Table 5.1, five focused properties of the busload can be modeled and detected by the means of DSPN. The criteria quantified in Chapter 4 can be also parameterized and, further if necessary, calibrated into the Petri net communication model.

So these behaviors need to be integrated and modeled with Petri net, in order to proceed with busload validation.

Table 5.1: Assignments for modeling

Characteristics	Quantitative measurement	Description means
Bus overload	Model-based detection, $U< [U_A]$	DLL, DSPN & rates
Burst	Message sequence detection	APL, DSPN & Rates
Message access mechanism	Model integration and simulation-based evaluation	DLL, DSPN & Rates
Bitwise arbitration	Model integration and simulation-based evaluation	DLL, DSPN & Rates
Message-sending occurrence	Goodness of fit	Log data & DSPN

As is shown in Figure 5.1, the communication model in this chapter consists of five Petri net sub-models in total. They are categorized and listed as follows:

- one channel state sub-model, initially shown in Figure 5.2 then extended structure shown in Figure 5.3.
- four message-source-nodes sub-models, 4th of which is shown in Figure 5.4.

5.2 DSPN Modeling Environment

The formal modeling with Petri nets is adopted in this approach due to its advantage of describing the communication protocols with rigorous mathematical and graphical definitions, see the book of Reisig [Reisig, 2012], compared with other formal methods. The motivation of applying Petri nets is mentioned in Section 2.2.3

Timed Petri nets play an important role in profiling the system dynamic properties. Especially for communication protocol validation, DSPN is hereby selected as modeling means.

The software π-*Tool* developed by the Institute of Traffic Safety and Automation Engineering, (iVA, das Institut für Verkerssicherheit und Automatisierungstechnik, Technische Universität Braunschweig, Braunschweig, Germany) has been built for the researchers who use Petri net as tools for RAMS analysis (Reliability, Availability, Maintainability and Safety) [Quiroga, L. M., Becker, U., and Schnieder, E., 2014]. It offers a comprehensive platform and a streamlined interface for creating comprehensive Petri nets models on diverse system hierarchical levels. All common stochastic distributions can be assigned to the relevant transitions, values of which are able to be read off. With the help of steady-state simulation and the transition firing rates generation of each individual transition, validation procedure in this work with the aspect of system performability can be further proceeded.

Typical stochastic distributions together with all deterministic time durations are available to set for transitions parameterization, the time definition of which indicates the waiting time of the transition triggered to fire. Hierarchical models are also provided in π-*Tool* with the concept of sub-model structure. Each sub-model can communicate with other sub-models by fusion places and diverse arc types.

In addition, the Petri net model can be further parameterized with transitions assigned by deterministic and stochastic timed quantities synthesizing the system model with diverse modeling characteristics. Stochastic distributions types, such as negative exponential distribution, uniform distribution, normal distribution, Log-normal distribution, Gamma distribution and Weibull distribution, are supported by the parameterization-assignment function in π-*Tool*. Therefore, four formerly mentioned types of stochastic distributions are applied in this work. Furthermore, Markov-Chain method is in this work excluded, because of the hybrid types of defined stochastic distributions and deterministic parameters assigned within the communication model.

The Monte-Carlo simulation method is adopted due to the system complexity and parameters of various stochastic and deterministic time involved. Instead of generating an exploded state space, Monte-Carlo simulation provides independent approaches than Markov-Chain method focusing on state reachability, which makes the model analysis less expensive and more efficient.

The previous iVA works related to Petri net and communication system can be found in Section 1.3.

5.3 Channel State Modeling by DSPN

In order to prevent ambiguous concepts before the modeling process begins, it is necessary to mention that the channel behavior is intuitively defined by two states. They are named bus free state and bus busy state. Bus free state implies that currently no message is transmitting on the channel. Alternatively, bus busy state indicates that at least one message is transmitting on the channel, *i.e.* the channel is in a working state at the moment. Both states are interpreted as two places, shown in Figure 5.2. The initial token inside the place "Bus free" indicates that the channel is available to be occupied.

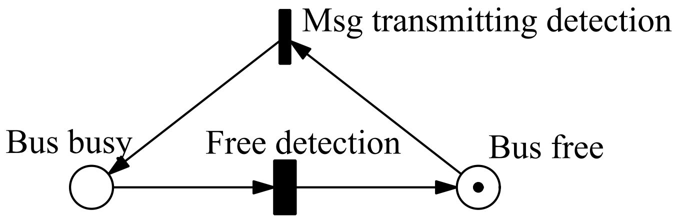


Figure 5.2: **Initial model of the channel state with Petri nets**

Two transitions here are introduced and connected with two places. Firstly, the transition "Free detection" is defined as a deterministic transition. The firing time interval in this transition is defined as one complete time unit of transmitting one message on the channel. This firing time depends on two key factors. They are the frame length and the bit rate. According to the time interval SmallCAN frame structure expressed by Equation 4.1 and Equation 4.2 mentioned in Section 4.3, it takes $7.1875ms$ to transmit one message in the SmallCAN fieldbus system. Therefore, the transition firing time interval is hereby represented as the time interval releasing the channel from a busy state back to a free state. The main Petri net elements in this sub-model are connected via normal arcs. It performs for either producing or consuming tokens in the local sub-model.

The token capacity of each place in this simple occupation model is set to one subject to the place attribute rule. Additionally, this avoids the improper channel state disobeyed by the serial communication rules, such as when both places are set with a token, leading to a dead lock. This implies the channel is transmitting the message at the same time it is still available to be occupied by the upcoming message, which violates the rule of serial communication in fieldbus. However, concurrent message-sending is inevitable in large scale fieldbus system. This is further discussed with the extended model structure in Figure 5.3. Another abnormal scenario that a token is consumed by extra Petri net elements results in no token existed at local resource. This leads to a vague local system state and need to be avoided. Both illogical scenarios mentioned above can be prevented by this Petri net modeling structure, two parts net structure of which is illustrated as follows:

Fusion Places: the process that messages from their sources occupy the channel can be mapped with a net binding structure between message-source-nodes model and the local channel state model. Therefore, the fusion places combined with the test arcs are hereby applied. The fusion places "Node 1. Bus 1", "Node 2. Bus 2", "Node 3. Bus 3" and "Node 4. Bus 4" representing this binding structure between these 4 message-source-node sub-model and the channel state sub-model. The tokens of these fused places are the outcome messages from the Medium Access net structure,

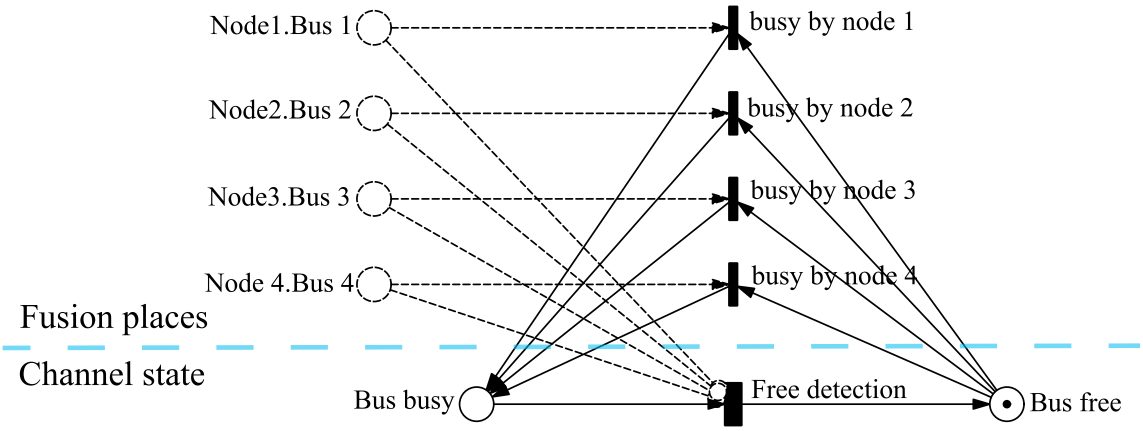


Figure 5.3: **Channel state sub-model with Petri nets**

Table 5.2: **Descriptions of net elements in channel state sub-model**

Name	Type	Description	Comment
Bus free	Place	Channel state currently holds no message and is available for any message to occupy.	Local
Bus busy	Place	State of the channel where at least one message is successfully sent to the bus channel, results in the state of the channel being currently unavailable for any node to occupy.	Local
Bus Node 1–4	Place	Places are fused with each message source sub-models' output onto the bus channel: these fusion places represent the functional mapping and the interactive control structure between the local channel state sub-net and other message source sub-models.	Fusion places
Free detection	Transition	Releasing the channel state from busy to free, minimum single message transmission time is necessarily set here as releasing time span.	T_S
Bus Busy by Node 1–4	Transition	The contextual resources controlled by fusion places from message source Petri net sub-models changes the channel state from bus busy to bus free, which can be seen as the extended version of transition "Msg transmitting detection" in Figure 5.2.	Local

which is further discussed in Section 5.4. **Channel state**: in order to change the global channel state to a busy state via triggering the transition "Msg transmitting detection", shown in Figure 5.2. However, according to serial communication, any pre-place fulfilled with token can trigger the channel state to bus busy. Therefore, the transition "Msg transmitting detection" needs to be restructured to four individual transitions connected to each fusion place via the test arc defined in Petri net rule, shown in Figure 5.3.

5.4 Message-Source-Nodes Modeling by DSPN

The message-source-nodes model concept is inspired by the work of [Schrom, 2003], which is constructed to introduced the local bus nodes integrated with CSMA/CA and bitwise arbitrary mechanisms. Model parameterization and simulation were not included.

In this work, the DSPN message-source-nodes model is built with complex fieldbus networking. The interconnection between each sub-model is also hierarchically implemented. Moreover, the model parameterization is based on quantitative analysis of the real message-sending behavior as well as the timed fieldbus protocol. Simulation and further analysis can be found in Chapter 6 and Chapter 7.

Definition 5.4.1. Message-Source-Nodes in this work is defined as the groups of messages identified by their assigned functions. Therefore, it is hereby explicitly represented as "message-source-nodes" instead of "single fieldbus bus node".

They generate messages with unique message IDs. The priorities of these four message-source-nodes are modeled on the predefined sequence in Figure 4.4 mentioned in Section 4.4. The channel state sub-model, which has already been introduced and extended above, is in charge of detecting and monitoring the traffic state on the channel. It represents the current traffic on the channel. In addition, the channel state sub-model is intercommunicated with four message-source-nodes sub-models via fusion places with different priority structures, shown in Figure 5.3. The descriptions of the channel state sub-model are shown in Table 5.2.

Due to serial communication topology, the moment one message successfully occupying the channel leads to the busy state. This behavior is constructed by four inhibitor arcs connected to the fused place from each message source. When it occurs that least one of these four fused places is fulfilled with one token, the test arc structure here would remain the local resource and requests to trigger relevant "Busy by node n" transition. This results in the global channel state switching the token from place "Bus free" to place "Bus busy". The further sending tokens must queue in the waiting places for further bus channel occupation.

In this chapter four independent message-source-nodes have been integrated as Petri net sub-models with similar net structures. The structural difference among these message-source-nodes sub-models are bitwise arbitration performed by different fusion places and thus representable. The 4th of them is shown in Figure 5.4 and especially described in Table 5.3.

In order to implement the net structure of the communication model, selected Petri net element types need to be introduced, which are normal arc, test arc and inhibitor arc.

A **normal arc** type is a basic element structuring other Petri net elements, it is in charge of passing the local token in order to change the local system state, in Figure 5.4, the message as token here generates from transition “Input 4”, stays at place “Send 4” and wait until the transition “Priority win 4” triggered, then the message begins to transmit on the bus channel to the designate receiver. All token path mentioned above is constructed via normal arc, in order to map the message behavior inside local source-nodes as message generation, access and transmission on the bus channel. In addition, any token generation or token consumption related Petri net structure is also constructed by normal arc type, such as the CSMA mechanism.

A **test arc** type here is involved because of conditional logic control. Extra fusion places from other sub-model play a part as the controlling context of the local sub-model, the tokens of which are hereby unconditionally held here. Such as these three test arcs between place “Node1. Bus 1”, “Node2. Bus 2”, “Node3.Bus 3” and transition “Priority lose 4 by node 1”, “Priority lose 4 by node 2”, “Priority lose 4 by node 3”. Any token in these three fusion places, represent a successful message occupying the bus channel with relative high priority. These net elements are applied in this work for implementing the interactiveness among each sub-nets. And the token in pre-place can not consumed by the test arcs during the token transmission. This preserves the token within the local sub-model while the conditional control can still be executed. For example, functions of executing the priority lose process is performed without losing their own tokens in the corresponding Petri net sub-models.

The test arc between message register Place “Send 4” and transition “Carrier Sense 4” in Figure 5.4 represents the token duplication for entering message CSMA access structure, the requesting message is still kept as local token at the sending register while the duplicated token represented as the logic control token penetrate the further message access part in this sub-model.

An **Inhibitor arc** type is selected for ensuring avoiding the abnormal token flow caused by concurrent transition. According to serial communication structure mentioned above, any fulfilled token in place “Node1. Bus 1”, “Node2. Bus 2”, “Node3.Bus 3” not only triggers the priority lose structure, but also disable the local transition “Priority win 4”. Therefore, each inhibitor arc need be involved pointing from any fusion place assigned with relative higher message priority to the transition “Priority win”, in order to prevent the chaotic transitions’ firing sequence, such as the concurrent transitions inside priority comparison part. Besides, the inhibitor arc pointing from place "Bus occupation. Bus busy" to transition "Carrier Sense 4" performs differently function, inhibitor arc blocks the entry of message entering the CSMA mechanism part the moment the bus channel state changed to bus busy.

In addition, **Fusion place** is applied in connecting each sub-model in a hierarchical way. Fusion places, such as the fusion places "Bus occupation. Bus busy", "Node 1. Bus 1", "Node 2. Bus 2", "Node 3. Bus 3" mentioned in Figure 5.3, Section **??** perform as the conditional control structure for bus state. For example, the place "Bus 4" of the channel state sub-model in Figure 5.4, is fused with the place "Node 4. Bus 4" of the message-source-node sub-model in Figure 5.3.

The local token flow inside this sub-model represents the message generation and transmission under the integrated message-access mechanisms. In order to explicitly illustrate the 4th message-source-node sub-model shown in Figure 5.4, which can be structured with the net sections as follows:

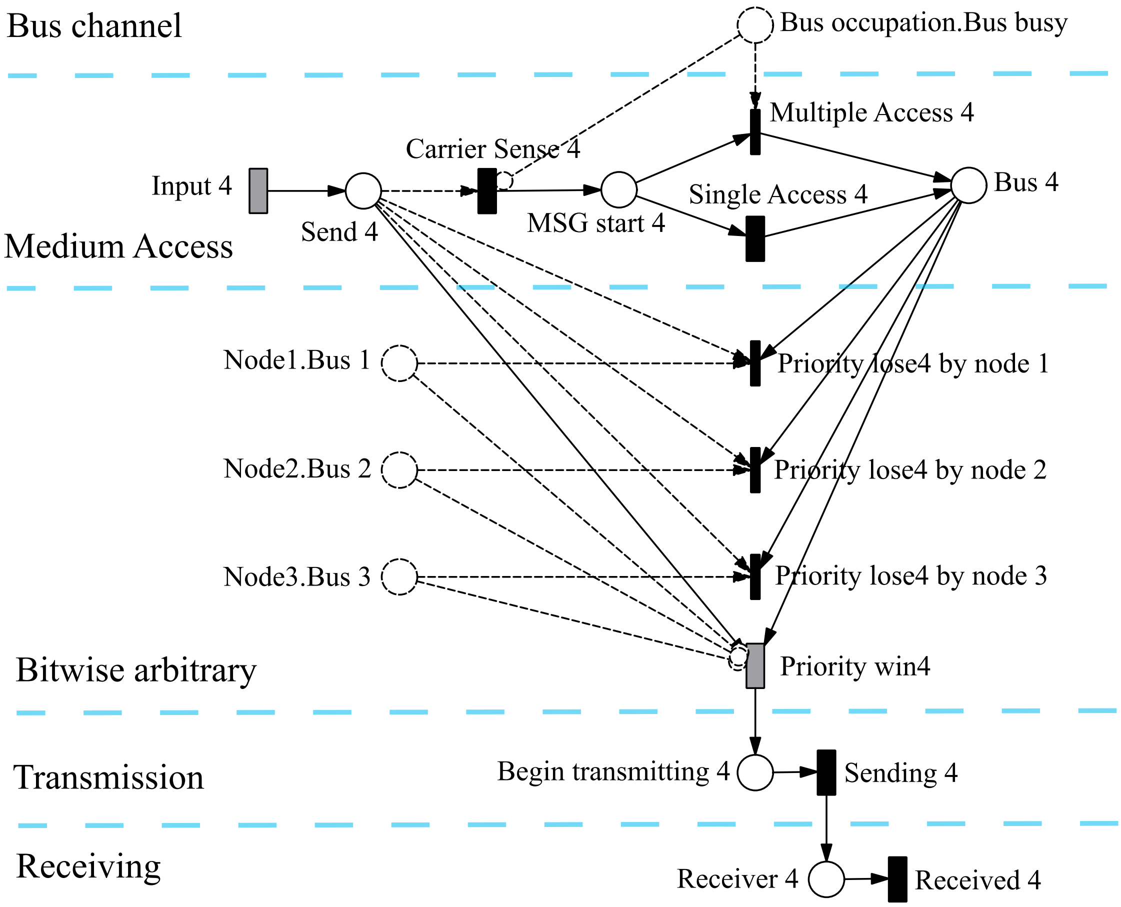


Figure 5.4: **4th message-source-node sub-model by Petri nets**

Table 5.3: Descriptions of net elements in the 4th message-source-node sub-model

Name	Type	Description	Time
Input 4	Transition	The message-sending frequency from the 4th message-source-node consisting part of the total throughput on the channel, parameterized by the fitted lognormal expectation.	μ_4
Send 4	Place	The token buffer. It duplicates the token for entering the sections of CSMA/CA and priority comparison. The token in this place can only be consumed by firing the transition "Priority win 4" when the message wins out and starts to send to the receiver.	
Carrier Sense 4	Transition	Entrance of the Medium Access structure. Only triggered when the token number both in register place "Send 4" is bigger than 1 and the fused place "Bus occupation.Bus busy" is 0.	
MSG start 4	Place	Two paralleled token paths are provided for the token in this place. One path need to wait a time interval defined in the transition "Sing Access 4" while the other path immediately triggers.	
Single Access 4	Transition	If the fused place "Bus occupation.Bus busy" remains empty, the token in the place "MSG start 4" will go this waiting path. This transition is parameterized with a short time interval according to Equation 4.3.	T_A
Multiple Access 4	Transition	If the fused place "Bus occupation.Bus busy" is set with a token, the token in the place "MSG start" will be brought into the place "Bus 4" by triggering this instant transition.	
Priority lose 4 by node 1 to 3	Transition	Connecting the fused places from their corresponding sub-models, remaining the buffering token in place "Input 4".	
Priority win 4	Transition	Any token set in the places "Node 1.Bus 1", "Node 2.Bus 2" and "Node 3.Bus 3" will immediately deactivate this transition with instant-firing transition subject to the bitwise arbitration.	T_A
Begin transmitting 4	Place	Hold the message-sending token went through the MAC structure for further transmitting	
Sending 4	Transition	Message transmitting process.	T_S
Receiver 4	Place	Hold the message-receiving token.	
Received 4	Transition	Consuming the received token.	T_R

Message generation section: transition “Input 4” and place “Send 4” consist of this. The timed parameter set in transition “Input 4” is mapped with as the time interval between every two message-sending events. The quantitative analysis, such as Sorting the log data from the real fieldbus system, is discussed in Section 4.5. Moreover, the quantitative analysis of fitting the stochastic distribution type of this time interval as a result of this log file sorting, is discussed in Section 4.6. As a result, the timed parameter defined in transition “Send 4” is assigned with the expectation and variance of the fitted stochastic distribution type, in this case, lognormal.

This type of transition “Input i”plays an important role for generating token flow with stochastic characteristic inside the Petri net communication model. This is mapped with the busload accumulation onto the channel. The stochastic characteristic is based on the interactions between the system and its environment, so the computerized simulation-based analysis of this stochastic sending is further discussed in Section 6.2.

Serial communication control section: transition “Carrier Sense 4” together with its pre-arc plays a key role of duplicating the generated message and then buffered and stored it as tokens in sending register represented as accumulated tokens in place “Send 4”. The goal of implementing the token duplication is to emphasize the message access section by means of MAC. In other words, the token flow in this Petri net module is mapped with MAC logic and bus channel occupation in global state whereas the token flow in the priority comparison module is mapped with the message consumption and bus channel release in global state. Moreover, concurrent message-sending is consistently prohibited according to general structure of serial communication. Therefore, the Petri net structure of one inhibitor arc from fusion place “Bus occupation. Bus busy” pointing towards transition “Carrier Sense 4” is consequently constructed in order to fulfill the specification of the serial communication. It indicates that further concurrent messages are prevented from entering the channel by the general bus busy state. This is done by any successful occupied message, disabling the token flow started from place “Send 4”.

CSMA/CA section: the prerequisite of token entering this Petri net structure is the global state of bus channel free, which means that no concurrent message-sending event exists. Due to the CSMA/CA mechanism, first net concurrency is hereby constructed. There are two ways for token transmitting from the place “Message start 4” to the place “Bus 4”: the first way of the medium access is to wait a fixed time delay defined in the transition “Single access 4” shown in Figure 5.4. The second way is to send the message instantly to the bus from the moment the global state of bus channel changed to bus busy, representing the detected concurrent-sending.

It is worthwhile to be mentioned that the concurrent net structure in this work is considerably important to mapping the concurrent message-sending. Therefore, the uncertain token flow generated concurrent net structure can be controlled in two ways in this Petri net modeling approach, which is summarized as follows:

- setting the post-transitions with different timed parameters.
- setting the post-transitions with different priority parameters.

by which the concurrency can be hereby avoided selecting one way to execute the token flow. However, the other token flow path is deactivated due to this net structure. As a result, it is necessary to include the extra Petri net elements, such as test arcs and fusion places, for recurrence the concurrency but under control.

The CSMA/CA section is suitable for the scenario with relatively high throughput involved and ready for the priority comparison among these concurrent-sending messages.

Besides, place "Bus 4" represents the bus channel and modeled within the local net structure. It means that bus channel would be paralleled occupied among these message-source-nodes, which obeys the serial communication rule. On the contrary, multiple channel occupancy in each section of the Petri net sub-models is only to profile the concurrent message-sending resolution. The real message transmission occurs in the transmitting and receiving sections.

The channel state would be possessed if any type of place "Bus i" is filled with token, *i.e.* any fulfilled local token in place "Bus i" will set the token to the global place "Bus occupation.Bus busy" which is still subject to the serial communication structure.

Bitwise arbitration section: Bitwise arbitration section is executed with the priority comparison. This would be initialized, when more than one message from various message-source-nodes are piling up before the actual occupation on the field-bus channel. As shown in Figure 5.4, all the fusion places "Node 1. Bus 1", "Node 2. Bus. 2" and "Node 3. Bus 3" are fused with their local place "Bus 1", "Bus 2" and "Bus 3" respectively, representing the dominant role of these local bus states with relative high priorities.

Any fulfilled token in these three fusion places "Node i. Bus i" would trigger its relevant post transition "Priority lose 4 by node i". This results in consuming the token in the place "Bus 4", releasing the channel busy state to free. Any concurrent transmitting event from messages with predefined higher priorities would trigger Multiple Access section and then enter the bitwise arbitration section.

This net concurrency is conditionally controlled by all the test arcs from place "Send 4" to transitions "Priority lose 4 by node i". The local transition "Priority win i" will be triggered by its message from the relevant message-source-node with the predefined highest priority. Therefore, one normal arc from the place "Send 4" is connected to the transition “Priority win 4”. Intuitively, the 1*st* message-source-node has no dominant fusion place because of its message source definition with the highest message priority. On the contrary, the last message-source-node, 4th sub-model node in this case, has been dominated by all the other fusion places, any other concurrent message would trigger its retransmission mechanism.

The CSMA/CA mechanisms as well as bitwise arbitration are integrated into these Petri net sub-models based on the quantitative analysis mentioned in Section 4. As a result, the Petri net model behaves with an precise approximation to the real MAC behavior in SmallCAN. This is implemented with combining fusion places and inhibitor arcs into the Petri net model. The fusion place mapped with place “Bus busy” in channel state sub-model has been built in each message-source-node in order to enable the corresponding token path by firing the transition“Multiple access”. The inhibitor arc, from the fusion place “Bus occupation” pointing to transition “Carrier Sense 4”, temporarily blocks the pre-token entering the CSMA mechanism, as can be seen in Figure 5.4 and Section 4.4. Two concurrent transitions have been set for different token paths for the channel access. If and only if the current channel state is busy, then the ongoing CSMA messages can be directly sent to the channel via transition “Multiple Access 4” for further priority comparison. Otherwise the relevant message has to pass through the transition “Single Access 4” with a defined waiting time. As a matter of fact, this CSMA structure has been equally established in each message-source-node sub-model.

Priority assignment of the fieldbus message sources is similarly designed to follow the rule defined in Figure 5.4 and Section 4.4. Complexity exists in implementing this model structure: if more fieldbus message-source-nodess are involved as Petri net sub-model for communication modeling, this phenomenon is especially obvious in the late stage that numbers of message sources are assigned with relatively low priority. The fact that more fusion places need to be involved in the Petri net model, leads to the consequence that integrated modeling elements for implementing this priority comparison structure are exponentially expanded.

Four message-source-nodes as Petri net sub-model are respectively constructed as the initial approach of modeling and parameterization. The message priorities in which are assigned with descending order, *i.e.* $P_1 > P_2 > P_3 > P_4$. In Figure 5.4, three fusion places “Node 1. Bus 1, Node 2.Bus 2, Node 3. Bus 3” with 6 test arcs are created here for interconnection part in priority comparison, means when the message collision occurs, the priority of this message-source-node has the lowest priority to occupy the bus channel among all message-source-nodes. The element descriptions of the 4th message-source-node sub-model are shown in Table 5.3.

5.5 Chapter Conclusion

This chapter has outlined the net structure and modeling assignments presented by attribute hierarchy in Section 5.1. Then the DSPN modeling environment together with its analytical features, in this work the software named π-Tool, is introduced in Section 5.2.

The concurrency handling mechanism, such as CSMA/CA and bitwise arbitration mentioned in Section 4.4, have been integrated into the hierarchical Petri net communication model. The model has been checked with token flow function, performing in a satisfactory manner under low traffic conditions.

According to the probability distribution fitting method mentioned in Section 2.3.4, the timed parameters of the message-sending frequency extracted and sorted from the log file of the fieldbus-based building automation system, shown in Section 4.6, are hereby parameterized with the transition firing expectations as model input. This is done in each message-source-node sub-model, as is shown in Section 5.4.

Based on this, the model is constructed with two kinds of sub-models, the channel state sub-model 5.3 and the message-source-node sub-model in Section 5.4. The interconnections together with the selected DSPN modeling elements are listed and explained in detail.

With the aim of approaching the model under the worst-case scenario, the method of which is mentioned in Section 2.3.5, the next steps of busload validation procedure can be enumerated as follows:

1. Involving more nodes in one Petri net communication model, providing relative heavy traffic density: one possible way of implementing this target is to apply the JAVA programming contributions to extend the communication model, as can been found in [Yiwen Chu, 2013]. The current maximum Petri net sub-nets can be accumulated to 200, providing adjustable large number message-source-nodes transmitting messages onto the bus channel. The further analysis based on this programming-based model extension is shown in Chapter 6.
2. Computerized methods, such as Monte-Carlo simulation method mentioned in Section 2.3.6 is applied in analyzing the Petri net communication model and validating the busload of the large scale fieldbus system SmallCAN. This is also the prerequisite of Chapter 6.
3. Function interrelations defined inside the Application Layer also have been mapped and extended into the Petri net model. Then, simulations and analysis of the transition firing rates, also have been carried out, shown in Chapter 7.

With more message-source-nodes sub-models involved in one channel state sub-model, the message-sending concurrency sharply rises. As a result, the state-space explosion cannot be avoided regarding net structure analysis. Another reason of not applying the reachability graph analysis is that the goal of this work is to validate the busload by analyzing the system dynamics. Therefore, the open net structure is hereby constructed by starting and ending in each message-source-nodes sub-model with a respective transition. By observing the selected transition-firing-sequence during the token flow, the correctness of model structure under the required assignments mentioned in Section 5.1 has been tested.

Chapter 6

Performability Evaluation of the Concurrent-Message-Sending

According to the system conceptualization mentioned in Chapter 3, concurrent-sending scenarios with selected access mechanisms have been classified and quantified in Chapter 4, which are further modeled and parameterized in Chapter 5.

In this dissertation, the concept of the performability is concerned with two aspects: performance and availability. As is shown in Figure 6.1, performability assignments are formally presented by the attribute hierarchy.

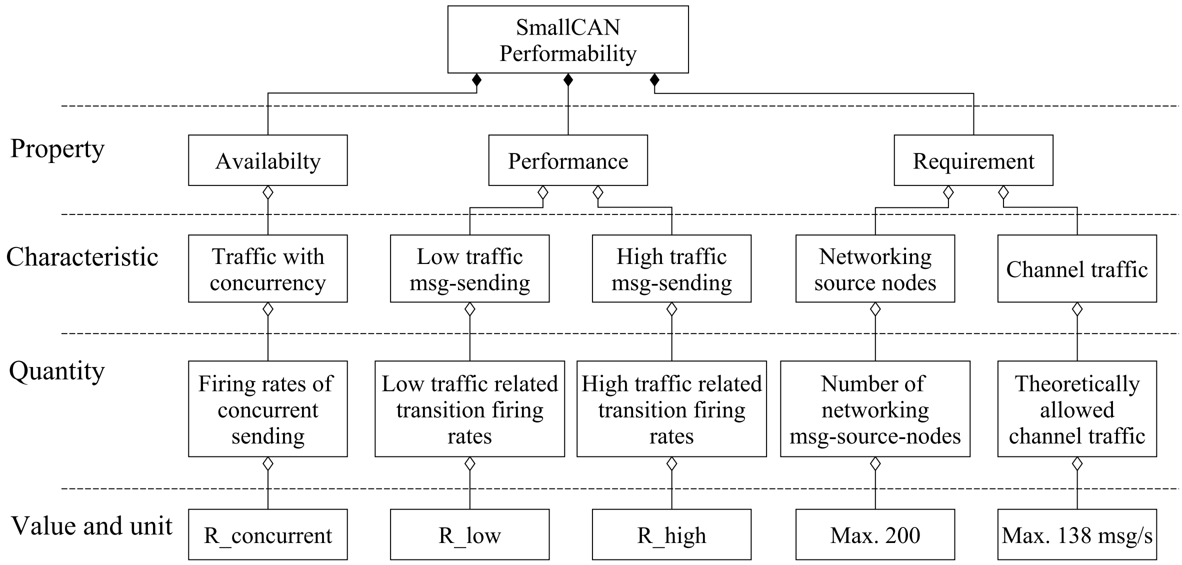


Figure 6.1: SmallCAN performability assignments by the attribute hierarchy

On one hand, availability analysis is concerning with the real-time traffic with concurrent-message-sending. The goal is to profile and track the allowed message arrival rate of the Petri net model under various concurrent conditions without impairing the function integrity.

On the other hand, performance evaluation is to describe and analyze the system behavior regarding the accumulating fieldbus complexity. It includes analyzing the busload with various traffic densities, evaluating the integrated access mechanism and interactions among message-source-nodes. The goal of this section is to analyze the model performance under the given constrains of computerized-modeling resources, varied up to the maximum achievable network structure.

Therefore, the performability of the fieldbus system can be evaluated by analyzing the computerized simulation-based results of the Petri net communication model.

The prerequisite of this chapter's work is based on the programming-based extension of the focused Petri net communication model. It is then followed by applying the computerized simulation method in order to generate the transition firing rates of the communication model. The focus of this chapter is on analyzing and evaluating the simulation results of sorted firing rates of observed transitions regarding the system performance and availability.

For this purpose, Section 6.1 starts with the availability analysis of the channel concurrency, following by a first approach of the Monte-Carlo-Simulation-based performance evaluation by low traffic density of 20 message-source-nodes sub-models. Then, is carried out with 200 message-source-nodes sub-models included, which are created by programming-based model extension. Based on this, performance evaluation under high traffic density is carried out with two aspects: the general system behavior in Section 6.3 and the local subsystem behavior, in this case, 20th message-source-nodes sub-model in Section 6.4.

The assumption is given that the function interrelations predefined inside the Application Layer are mapped within groups of messages. Each group of messages controlling or executing the same function assignments is independent with each other. Then, they are defined as message-source-nodes. Therefore, the total message-sending frequency can be interpreted to the algebraic sum of sending frequencies of the message-source-nodes.

6.1 Availability Analysis of the Channel Model

The prerequisite of availability analysis is to focus on abstracting the Petri net fieldbus channel state model with optimized function and structure, as shown in Figure 6.2.

The real-time availability of the bus channel is calculated in Equation 6.1,

$$Avail_{\_Throughput} = \frac{R_{\_Msg\_Consuming}}{R_{Msg\_Arrival\_Mean}} \tag{6.1}$$

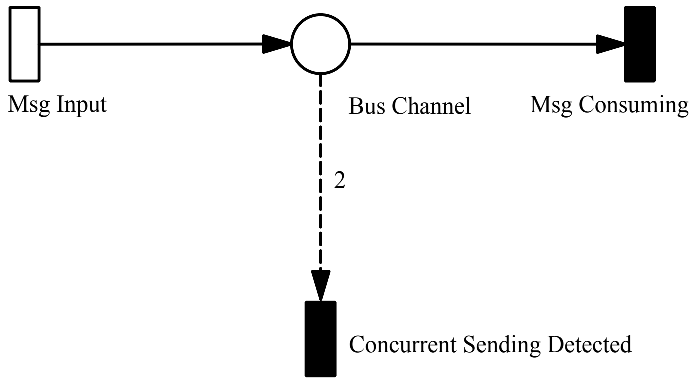


Figure 6.2: **Concurrent sending detection model by Petri nets**

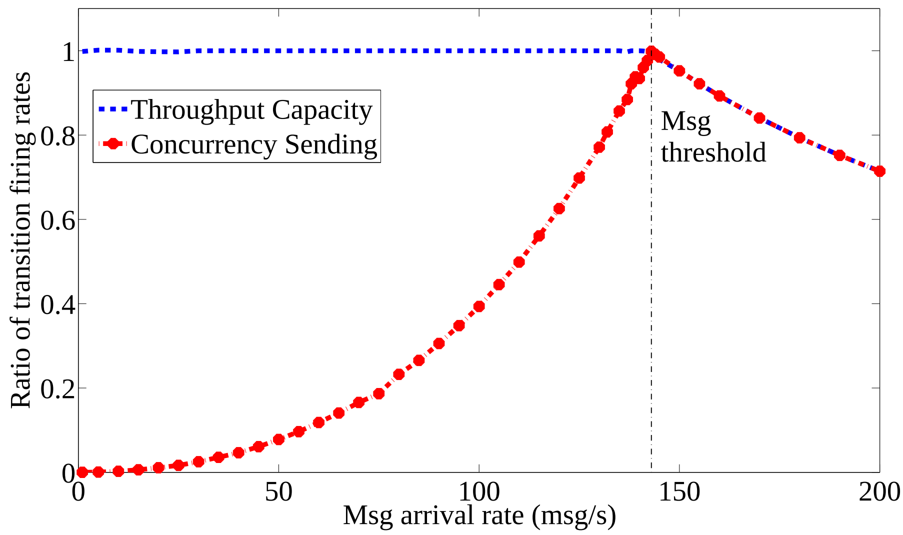


Figure 6.3: **Availability analysis of the concurrent-sending-detection model**

$R_{\_Msg\_Arrival\_Mean}$ is transition rate parameterized as exponential expectation value into the transition "Msg Input". It implies the average time interval among generated messages to occupy the bus channel. $R_{\_Msg\_Arrival\_Mean}$ represents the firing rate of the observed transition "Msg Consuming" deduced from the Monte-Carlo simulation results. Therefore, the quotient between these two quantities usually should be equal to 1 under the normal traffic state. The probability of the concurrent transmissions is calculated with Equation 6.2.

$$P_{\_Concurrency\_Sending} = \frac{R_{\_Concurrent\_Sending\_Detected}}{R_{Msg\_Arrival}} \tag{6.2}$$

Similarly, $R_{\_Concurrent\_Sending\_Detected}$ denotes the firing rate of the observed instant transition "Concurrent Sending Detected". It signifies the concurrent message-sending frequency due to the currently defined message releasing time span.

On the one hand, the channel throughput curve in Figure 6.3 indicates that the bus channel is feasible to hold the current message-sending frequency under the ordinary message arrival rate. On the other hand, with an increase in the mean value of the message arrival rate, the concurrency sending curve shows that the concurrent-sending frequency also rises exponentially. After the upper limit of the message-sending frequency has increased to 137 msg/s, the maximum channel traffic has been reached. Both curves then keep their original trends until reaching to the theoretical threshold of the message-sending frequency. It indicates that more messages are piling up on the bus channel waiting for occupation. Therefore, concurrent sending events have thereby constantly occurred on the channel. So the system degrading state is shown in the latter stage that the trajectories of both curves are merged together, and the channel traffic is intuitively and continuously declining.

This case study of transition firing analysis has further validated the threshold of the total message-sending frequency on the bus channel, which is also mentioned in the dissertation of Schrome [Schrom, 2003]. The result of this validation approach can be further quantified into the fieldbus protocol of the relevant system.

6.2 Performance Evaluation of the Extended Model under Low Traffic

The communication model mentioned in Chapter 5 has been extended in a programming-based manner, the maximum number of message-source-nodes integrated into one fieldbus system can be up to 200. In this section, first the model size is composed with 20 message-source-nodes Petri net sub-models and bus state sub-Petri net model. In the following Section 6.3 the model size will be extended to 200.

6.2.1 Timed Parameterization of the Extended Communication Model

Figure 6.4 shows the parameterization structure between the sorted message-sending occurrence in reality and the hierarchical Petri net communication model.

These two types of sub-models shown in gray in Figure 6.4 are message-source-nodes sub-model in Figure 5.4 in Section 5.4 and channel state sub-model in Figure 5.3 in Section 5.3. By sorting event-based message-sending types mentioned in Section 4.4, timed parameters are set into each message-source-nodes respectively, generating messages onto the bus channel.

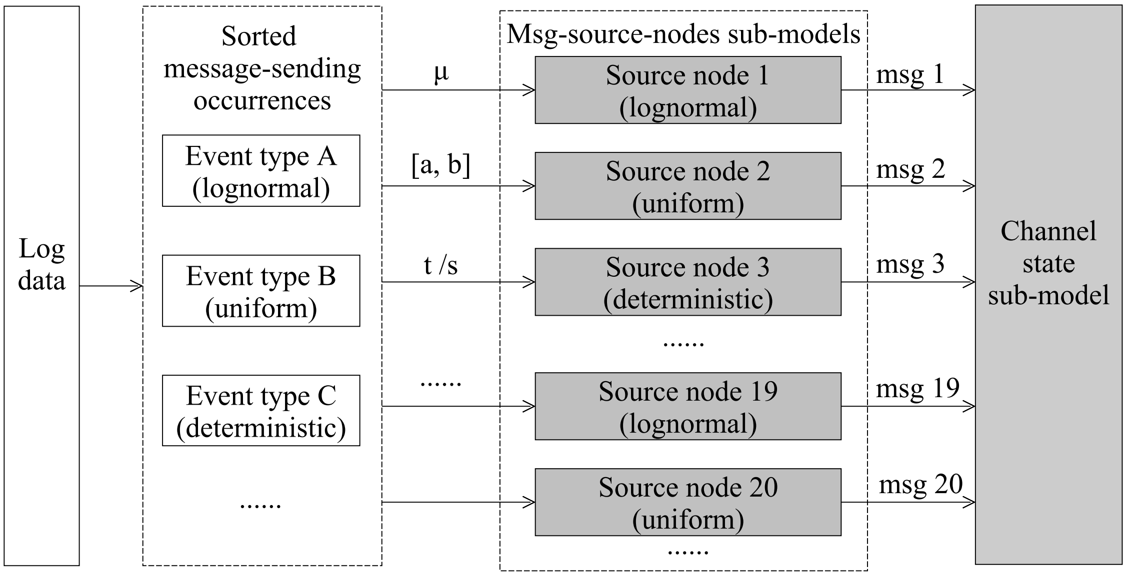


Figure 6.4: **Parameterization structure of the flexible model extension**

After the extended communication model has been constructed and parameterized, shown in Figure 6.4, it is essential to incorporate the parameters of the timed behavior into the model. Therefore, the timed behavior of the extended communication model can be categorized into two categories. They are the system contextual relevant timed behavior and the fieldbus protocol relevant timed behavior.

On one hand, the contextual relevant timed behavior of the extended communication model is deduced from the environment of the implemented fieldbus system. It can be described as the sporadic and periodic message-sending behavior triggered from the system context. It is also calculated as the deterministic or stochastic parameter set as model input of each local fieldbus message sources. As is shown in Figure 5.4, the former is represented as the time delay parameterized in transition “Input 4”. The transition firing delay is also assigned with the time interval of sorted message-sending occurrences, shown in Figure 6.4.

According to the quantitative analysis based on the SmallCAN specifications mentioned in Section 4.3 and Section 4.4, the fieldbus protocol relevant timed behavior is parameterized with the same sub-model timed structure mentioned in Figure 5.4 in Section 5.4 and Figure 5.3 in Section 5.3.

In this section, under relative low traffic density according to the stochastic fitting approach mentioned in Section 4.1, the timed input rule of each twenty message-source-nodes is parameterized by Equation 6.3.

$$T_{inputs} = \begin{cases} \mu & 3n-2,\ 0 < n \leq 7,\ n \subseteq \mathbb{N},\ lognormal \\ [a,b] & 3n-1,\ 0 < n \leq 7,\ n \subseteq \mathbb{N},\ uniform \\ 1s & 3n, \quad\quad 0 < n \leq 7,\ n \subseteq \mathbb{N},\ deterministic \end{cases} \tag{6.3}$$

Table 6.1: Attributes of temporal bus behavior regarding the SmallCAN protocol

Property	Characteristic	Quantity	Value & Unit	Symbol
Decentral function	CSMA/CA	Time interval in single access	1ms	T_access
	Bitwise arbitration	Priority comparing time defined due to its frame's arbitration field	uniform	T_win
	Message-sending	Time interval of transmitting message after successful channel occupation	20ms	T_trans
Central state	Channel releasing	Deterministic time interval of releasing bus state	7.1875ms	$T\_release$

The reciprocal of μ is the expectation of lognormal distribution and $[a, b]$ denotes the uniform distribution, these stochastic distributions are sorted and fitted by the relevant message-sending types from system log data in reality. It indicates the contextual influence on the fieldbus system, which contributes to system complexity in a stochastic or deterministic way. However, the message-sending behavior is independent from the numbers of networking fieldbus message-source-nodes. It is also irrelevant from the timed behavior defined in fieldbus protocol. Therefore, based on more numbers of the message-source-nodes with contextual parameters incorporated, the total message-sending frequencies can be algebraically accumulated. The busload on channel can thus be categorized into relatively low and high traffic density.

On the other hand, the fieldbus protocol relevant timed behavior is specified in Table 6.1 by the attribute hierarchy. The local state in the first column represents the timed scenario from the message-source-node while the global state means the timed behavior on the bus channel. It is worth identifying that the protocol-relevant timed parameterizations of local states are empirically defined for experimental data analysis.

This transition rates analysis approach is based on the Monte-Carlo simulation results. The Petri net communication model is composed with one channel state sub-model and 20 message-source-nodes sub-models implemented in π-*Tool*. Up to 686 transitions' firing rates are hereby generated and sorted for this analysis approach. So the evaluation of the transition firing rates are thoroughly discussed in following sections of this Chapter.

6.2.2 Bus Channel Occupation vs. Sending Frequency

The goal of this section is to validate whether the bus channel can tolerate and transmit the current messages, even if losing message due to concurrent sending. This work conducted the comparison of the firing rates between the transition "bus channel occupation" and the transition "Sending frequency". The current throughput has been determined from the aspect of the bus channel. Therefore, deterministic input frequency is parameterized into the models with message-source-nodes number $(3n - 1, 0 < n \leq 7, n \subseteq \mathbb{N})$ for simulation.

The black curve in Figure 6.5 shows the occupancy rate of the message-source-nodes with deterministic sending frequency $(3n - 1, 0 < n \leq 7, n \subseteq \mathbb{N})$. It performs an accelerated increasing tendency. The reason of this abnormal phenomenon is the same time interval of the deterministic model inputs of local message-source-nodes. Therefore, the more such parameterization with the same time interval are involved as the deterministic model inputs, the worse abnormal behaviors reflected by the oscillatory peaks in the latter phase of the black curve.

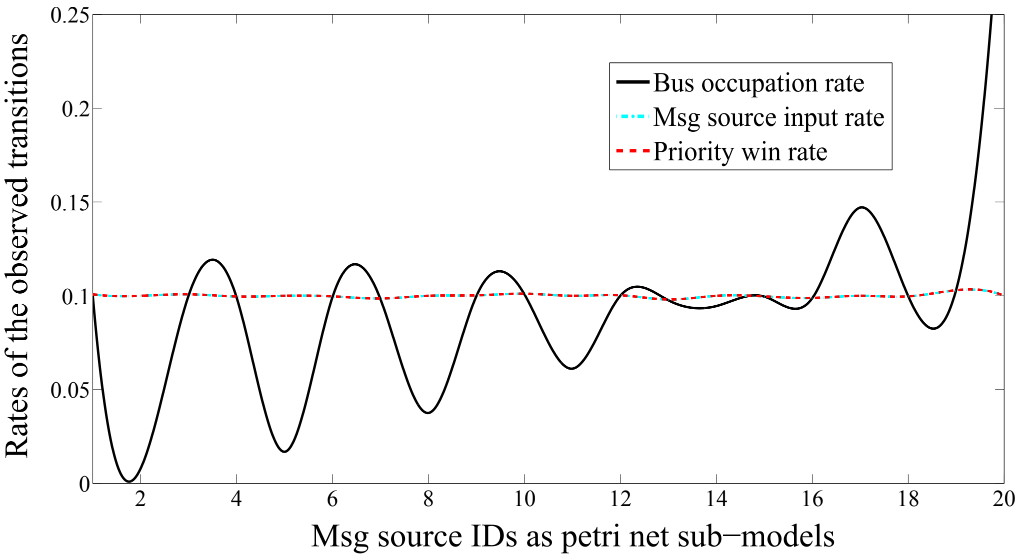


Figure 6.5: **Performance of source-nodes with stochastic and deterministic input**

As is shown in Table 6.2, message-source-nodes are listed with descending priorities. On the contrary, the blue curve is composed of the contextual model inputs with lognormal distribution as message-sending frequency has successfully held the current traffic.

In this case study, the deterministic time intervals of the input transitions of message-sending occurrences are defined as 1 second. Therefore, with large numbers of simulation steps of the Monte-Carlo method, accumulative peaks of all occupied messages with deterministic sending time interval have occurred at this time interval. On the contrary, the other message-source-nodes have stochastically distributed their sending rates over the simulation time scale. The overlapping between red curve and light blue curve shows that the current message traffic composed of message-sending frequencies' sum can be held by the channel.

In order to prevent the abnormal state mentioned above, deterministic message-sending frequencies are replaced by stochastic distributions. They are parameterized into all local message-source-nodes, including processes, such as message-sending and channel occupation. The clear comparison between these two parameterization has been concluded and shown in Table 6.2. The $\lambda_{\_Arrival\_cap}$ means the capability of bus channel throughput, which is calculated in Equation 6.4.

Table 6.2: **Transitions firing rates analysis of bus occupy and message source input with and without deterministic message-sending frequency.**

Node with descending priority	Bus occupy rate (before**)	Node input rate (before)	Ratio (before)	Bus occupy rate (after**)	Node input rate (after)	Ratio (after)
1	0.100156	0.100765	0.993956	0.100496	0.100576	0.999205
2	0.000294	0.099989	0.002944	0.019834	0.019855	0.998942
3	0.100232	0.100767	0.994691	0.100059	0.100119	0.999401
4	0.099048	0.099599	0.994468	0.100731	0.100788	0.999435
5	0.016849	0.099989	0.168507	0.020013	0.019981	1.001607
6	0.099416	0.099675	0.997411	0.100086	0.100122	0.99964
7	0.098098	0.098581	0.995094	0.10231	0.102368	0.999433
8	0.037547	0.099989	0.375516	0.019659	0.019650	1.000438
9	0.099909	0.100224	0.99686	0.099867	0.099886	0.999806
10	0.100764	0.101096	0.99672	0.100001	0.099989	1.000125
11	0.061153	0.099989	0.6116	0.019455	0.019461	0.999692
12	0.100039	0.100261	0.99779	0.100155	0.100113	1.00042
13	0.097618	0.098026	0.99584	0.099053	0.098891	1.001632
14	0.094525	0.099989	0.94535	0.019948	0.019963	0.999259
15	0.099581	0.099672	0.99908	0.100021	0.099996	1.000255
16	0.098735	0.098801	0.99933	0.100555	0.100465	1.000896
17	0.147033	0.099989	1.47049	0.019328	0.019307	1.001119
18	0.099842	0.099684	1.00159	0.100219	0.100143	1.000759
19	0.103234	0.102868	1.00356	0.100888	0.100688	1.001986
20	0.346589	0.099989	3.46628	0.01971	0.01969	1.001021
Sum	2.000664	1.999942	1.00036	1.442387	1.442049	1.000234

Note:
* The nodes with $(3n-1, 0 < n \leq 7, n \subseteq \mathbb{N})$ are assigned with the identical deterministic message-sending frequency in the parameterization called "before".
** The only difference between "before" and "after" is the homogeneous stochastic input parameterization with the nodes number of $(3n-1, 0 < n \leq 7, n \subseteq \mathbb{N})$.

$$\lambda_{\_Arrival\_cap} = R_{\_Bus\_occupy}/R_{\_Node\_input} \tag{6.4}$$

The comparative result between column “Ratio before” and column “Ratio after”, especially in row 17 and row 20, reveals that the abnormal state has been eliminated and indicates a homogeneous message-sending environment of stochastic distributions.

It is worth indicating that the reason of the slight difference between each rates of column “Bus occupy rate” and column “Node input rate” is due to the multiple access access mechanism implemented in inside each message-source-node model. The inhibitor arc between the fused place ”Bus occupation.Bus busy” and the transition ”Carrier Sense 10” in Figure 6.6 can prevent the further sending from the buffer register of each message-source-nodes. However, any message already begin to wait for a single access process can be immediately allowed to send to the bus channel for further comparison, resulting in the fact that not all sending requesting to the bus channel are via bus occupation. Because no matter which message is sent out, each bus occupation is only authorized to one successfully sending message. Therefore, due to Table 6.2, under low traffic density, the message lost scenario mostly cannot be found. And the ratio between bus occupy rate and message source input rate means that 99% of bus messages on average generated from each message-source-node can be occupied the bus channel for further access mechanism.

6.2.3 Single Access vs. Multiple Access

As mentioned above, before the generated message from each message-source-node requests the occupation state, the medium access mechanism is triggered and arbitrate all the requesting messages, whether short waitings should be set in the Petri net sub-model. It performs as an key role as one part of access structure in each message-source-node handing concurrency. $1ms$ is set to occupy the channel for single access to bring the requested message on the free bus channel when spontaneous sending messages are detected, whereas multiple access let the message directly send to the bus channel access without waiting. Therefore, the instant transition is set for representing immediate completion of this access process.

The firing rate of transition ”Multiple Access” or the ratio between single access and multiple access indicate the traffic concurrency. Therefore, it is necessary to compare the transition firing rate related to access mechanism for each Petri net sub-model. The Single / Multiple Access comparison is listed in Table 6.3.

The higher rate of multiple access of this local nodes is, the higher traffic other local node generates. Therefore, we can see that node number with $(3n - 1, 0 < n \leq$

Table 6.3: **Transition firing rates analysis of single/ multiple access mechanism**

Node	Single access	Multiple access	Ratio
1	0.100496	0.000137764	0.998631036
2	0.0198339	5.12E-05	0.997425208
3	0.100059	0.000119504	0.998807089
4	0.100731	0.000146713	0.998545635
5	0.0200126	4.03E-05	0.997990316
6	0.100086	0.000137677	0.998626303
7	0.10231	0.000146311	0.998571967
8	0.0196589	5.10E-05	0.997412468
9	0.0998665	0.00012158	0.998784055
10	0.100001	0.000230341	0.997701906
11	0.0194553	7.76E-05	0.996027216
12	0.100155	0.000146222	0.998542171
13	0.0990525	0.000196137	0.998023781
14	0.0199477	4.63E-05	0.997684305
15	0.100021	0.000156058	0.998442178
16	0.100555	0.000182514	0.998188222
17	0.0193284	6.93E-05	0.996427411
18	0.100219	0.000146769	0.998537659
19	0.100888	0.000105022	0.998960106
20	0.0197099	6.10E-05	0.996914657

$7, n \subseteq \mathbb{N}$) with lower transition rate, which means that these message-source-nodes have less impact on the bus traffic as well as the current traffic density level. This indicates that other message-source-nodes is also low. Because among nearly each transmitting event these requested messages have occupied the bus channel by means of single access without so many collisions occurred. The reason of all single access rates of the message-source-nodes ($3n - 1, 0 < n \leq 7, n \subseteq \mathbb{N}$) is much lower is that the relatively larger scale normal distribution duration defined in the input sending time interval of the local nodes numbers of ($3n - 1, 0 < n \leq 7, n \subseteq \mathbb{N}$), while the rest local nodes have 5 times quicker sending frequency. From this point of view, 20 message-source-nodes with current sending frequency definition, the traffic continues to be quite low. However, collision scenario still occurs especially in the local nodes with lower priorities defined.

6.2.4 Collision Scenario

According to the purpose of this modeling approach, the worst-case scenario (*i.e.* collision scenario), can be reviewed by data analysis of Monte-Carlo simulation results. Any triggered priority comparison mechanism indicates one conflict resolution. Therefore, it is necessary to have a thorough analysis and discussion in this part.

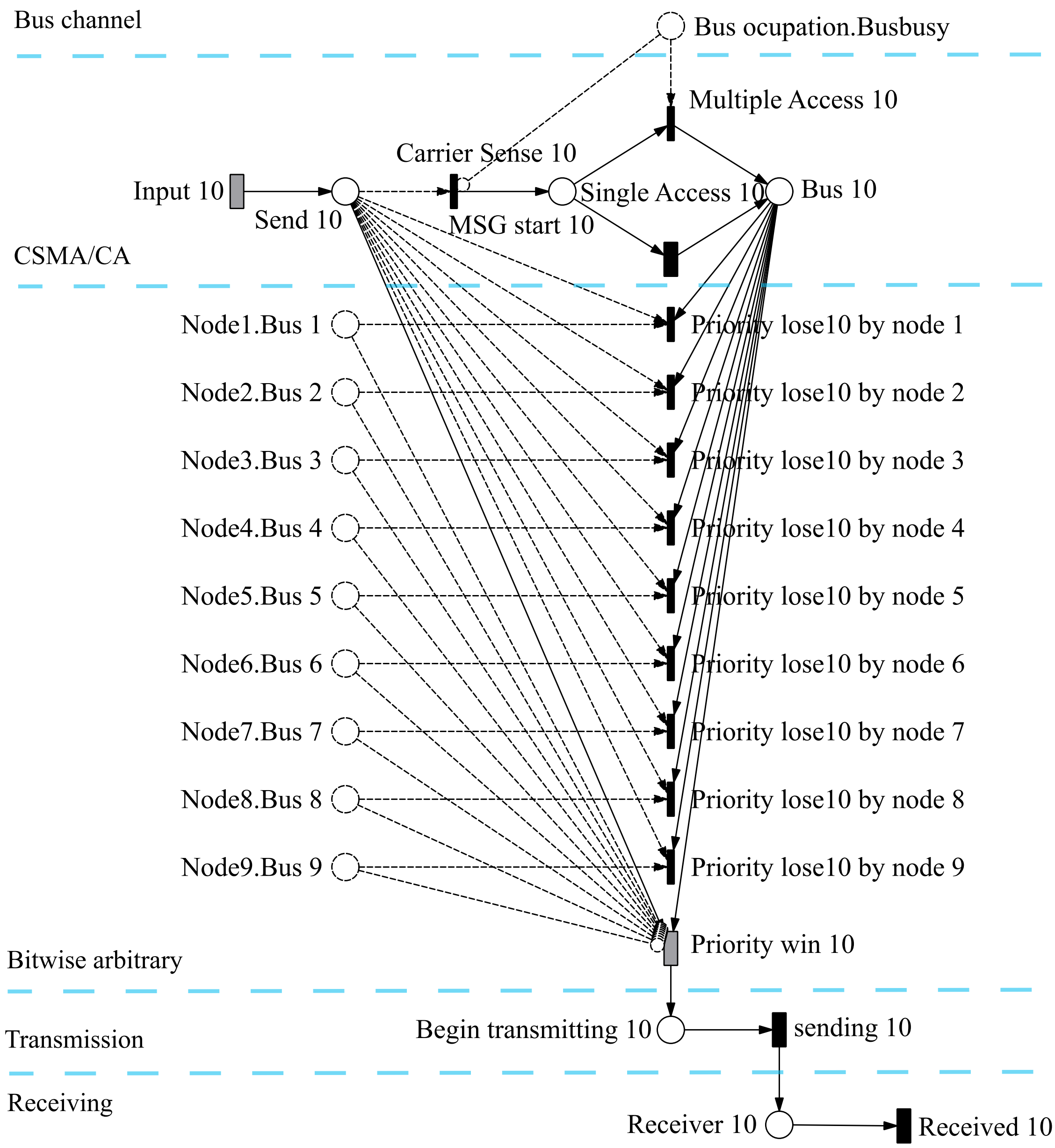


Figure 6.6: **Extended Petri net sub-model of 10th message-source-node**

According to each Petri net sub-model structure, there are only two consequences when a generated message completes the single/multiple access mechanism and accomplishes the priority comparison. These two consequences are simply named priority win and priority lose. Each consequence is decided by the already defined address part inside each message. And it is recordable and traceable that node-generated message triggers the transition "Priority lose" at each Petri net sub-model representing each message source, *i.e.* bus node.

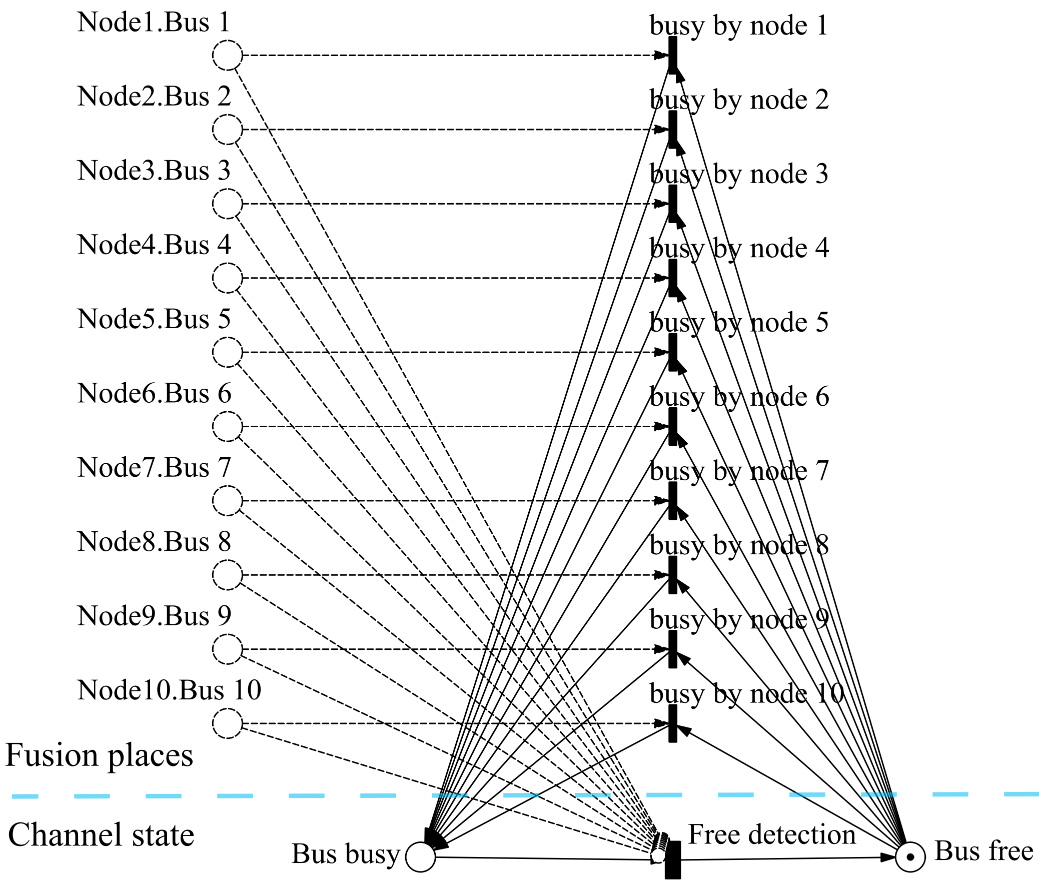


Figure 6.7: **Extended Petri net sub-model of channel state**

On the bus channel's aspect, it turns out that the firing rate of transition "Priority win" occurs much less than the firing rate sum of transition "Piority lose" in each Petri net sub-model under the background of the channel traffic generated from homogeneous message-sending occurrences.

As a case study for priority lose analysis is shown in Figure 6.6, there are 9 transitions "Priority lose 10 by node 1, by node 2, ... by node 9" in charge of priority lose mechanism designed in node 10, the pre-places of these nine transitions are fusion places connected with bus channels from all the other nodes, the priorities of which are respectively defined higher than node 10. So the configuration of message priority structure is mapped into each of the Petri net sub-model.

In order to thoroughly analyze the transition firing rate of priority comparison mechanism in message collision scenario, it is necessary that the firing rates of transitions "Priority lose" are comprehensively categorized into two sets, which are defined by Equation 6.5 and 6.6 respectively.

$$R_I = \sum_{j=1}^{j=i-1} r_{i,j}, \{r_{i,j} | \; i = Constant; j < i; r_{i,j} \in r_{Priority\_lose_i\_by\_node_j}\} \tag{6.5}$$

$$R_J = \sum_{i=j+1}^{i=n_{max}} r_{i,j}, \{r_{i,j} | \; j = Constant; j < i; r_{i,j} \in r_{Priority\_lose_i\_by\_node_j}\} \tag{6.6}$$

A case study is hereby presented. "$Priority\_lose_i\_by\_node_j$" is defined among these nine transitions of "Priority lose" functions in Figure 6.6 and analyzed with two variables. The variables "i" and "j" represent the number of implemented Petri net sub-models as message source trying to occupy the bus channel as well. The detailed introduction of the variables "i" and "j" are explained as follows.

- i in Figure 6.6 is 10, representing the 10th message-source-node.
- j' is the number of other message-source-nodes sub-models, *i.e.* 1st to 9th message-source-node with higher priority than 10th message-source-node 10.
- $i > j$ due to the descending listing priorities defined in the nodes' models.

If any collision occurs between messages generated from these message-source-nodes number $j \in \{1, 2, 3, ..., 9\}$ and node number $i = 10$, at least one token will be set among these fused places. As a result, the inhibitor arcs between the token-set fusion place and "Priority win 10" prevents the message from message-source-node 10 to win out during the collision scenario.

Table 6.4 shows the transition firing rates regarding two sorting ways of priority comparison. Priority win is intuitively sorted only in the second column. It indicates the influence degree on the channel occupation.

The first set of analyzing the transition firing rate of "Priority lose" shown in Equation 6.5 are sorted with the aspect of the fixed number i of this message-source-node. The sum of each firing rate of transition "$Priority\_lose_i\_by\_node_j$" in the Petri net sub-model number i represents the importance and the probability of bus channel occupation in node i during the message collision scenario.

Each value "1st rate sum" in third column of Table 6.4 is calculated by Equation 6.5. The rate sum R_I indicates the importance of this message-source-node i as message generating source to the bus channel, *i.e.* the Petri net sub-model as one message-source-node, the ratio between its generated message successfully occupy to bus channel and lost when comparing the priority with other message-source-nodes during the collision.

The second set is illustrated by the Equation 6.6 is sorted the firing rate sum of transition "$Priority\_lose_i\_by\_node_j$" on the basis of fixed "j". R_J shows the relative rate of the message-source-node "j" muting other nodes with the aspect of bus channel. Each value "2nd rate sum" in the fourth column in Table 6.4 is calculated by Equation 6.6, R_J indicates the traffic density of the current collision impacted by node j on the bus channel.

Table 6.4: **Transition firing rates analysis of priority comparison mechanism**

Node	Priority win	1st rate sum	2nd rate sum
1	0.100608	0	5.03E-04
2	0.0198716	1.25E-05	2.02E-04
3	0.100135	6.50E-05	0.001779
4	0.100806	9.12E-05	0.000774
5	0.0200001	4.39E-05	8.07E-05
6	1.00E-01	0.000126749	0.00047976
7	0.102385	0.000137731	0.000290302
8	0.0196589	6.50E-05	0.000428645
9	9.99E-02	0.001569996	0.003894537
10	0.1	0.000416236	0.000192859
11	0.0194804	7.26E-05	9.70E-05
12	0.10013	0.003868728	0.000246313
13	0.0989397	0.000658014	0.000209199
14	0.0199728	4.19E-05	0.000202785
15	0.100008	0.000465273	0.000190956
16	0.100468	0.000550658	0.000166076
17	0.0193284	0.000292074	6.97E-05
18	0.100157	0.000609407	8.19E-05
19	0.1007	0.00069734	0
20	0.0196973	0.000116877	0

These two sets of sorting the firing rate sum are arranged for analyzing transitions "Priority lose" in the collision scenario with the aspects of local subsystem and global system respectively.

Based on the prerequisite of rather homogeneous sending frequency distribution as input of the Petri net model, Table 6.4 shows message-source-nodes number ($3n-1, 0 < n \leq 7, n \subseteq \mathbb{N}$) with slower sending frequency. Less transition firing rate "Priority win" results in less impact and involvement from this nodes' sort during collision scenario. The case study mentioned in Figure 6.6 shows that the calculated firing rate sum 0.000416236 of these nine transitions "Priority lose 10 by node 1, ..., Priority lose 10 by node 9" result in the tenth row, third column of Table 6.4. The first zero in third column at node 10 implies that message from this node has the highest priority and it is impossible to lose during the priority comparison scenario. Whereas the last two zeros in the fourth column indicates that node 19 and 20 have no impact on the bus channel when a collision occurs between their messages and other nodes' messages.

As a part of the Petri net model, 20 priority structure has been implemented and parametrized. This is reasonably abstracted from the fieldbus communication protocol, such as the system failure report message as one of the periodic system state messages are provided with relative higher priorities than the normal function control messages defined in the messages' address sections.

Therefore, based on the analysis of the transition firing rates in the collision scenario, the formal results of this performance evaluation provide criteria for completing the fieldbus protocol and system implementation.

6.3 Performance Evaluation of the Extended Model under High Traffic

On one hands, according to the message-source-node sub-model shown Figure 6.6, the net elements in each model extension can be categorized into two sets: fixed net elements and variable net elements. The number of fixed net elements are constant, including 6 places, 7 transitions and 14 arcs. This is a total of 27 fixed elements in each message-source-node sub-model. The variable net elements are ascribed to constructing the structure of bitwise arbitration, which are listed as 9 transitions "priority lose 10 by node 1" to "priority lose 10 by node 9", and 9 corresponding fusion places "Node 1. Bus 1" to "Node 9. Bus 9", which are connected by 27 arcs. This is a total of 45 elements. Therefore, if the adjusted number of extended message-source-nodes is n, where $n \in \{n|n \geqslant 2\}$, variable elements in each message-source-node sub-model can be calculated as $5(n-1)$ whereas total net elements contributed by all message-source-nodes sub-models is calculated in Equation 6.7.

$$N_{source-nodes}(n) = 5\left[(n-1) + (n-2) + ... + 1\right] + 27n \tag{6.7a}$$

$$= \frac{5(n-1)(n-2)}{2} + 27n \tag{6.7b}$$

On the other hands, a case study of the channel state sub-model integrating 10 message-source-nodes is shown in Figure 6.7. The channel state sub-model has 5 fixed net elements, which are listed as two places "Bus busy", "Busy free", one transition

"Free detection" and two arcs. The variable net elements are 10 fusion places "Node 1. Bus 1" to "Node 10. Bus 10" and 10 corresponding transitions "busy by node 1" to "busy by node 10". They are connected by 40 arcs. Therefore, the sum of total net elements contributed by channel state sub-model is calculated in Equation 6.8.

$$N_{channel-state}(n) = 6n + 5 \tag{6.8}$$

As a result, the total Petri net elements under adjustable extension nodes number n is calculated in Equation 6.9

$$N_{total}(n) = N_{source-nodes}(n) + N_{channel-state}(n) \tag{6.9a}$$
$$= \frac{5(n-1)(n-2)}{2} + 33n + 5 \tag{6.9b}$$

It is concluded that the total Petri net elements involved in model extension has a polynomial growth rate to the adjustable number of message-source-nodes n.

In this section, the extended communication model consists of 200 message-source-nodes sub-models and channel state sub-model. This number is adjusted based on the maximum power of handling the exponentially increasing net elements by the current available computer resources. It can be improved in the future although the computer handling capacity reaches.

This section focuses on analyzing the worst-case scenario of the communication model with more nodes providing relative high traffic density on the bus channel. Worst-case scenario in this work is quantified by the maximum number of networking message-source-nodes with contextual input into the model under the limited computer conditions. Therefore, each Monte-Carlo Simulation approach with corresponding extended Petri net communication model has been conducted to generate transition firing rates.

According to the parameterization mentioned in Section 6.2.1, the relative high fieldbus traffic density has been illustrated with two aspects. First, one fieldbus with large scale numbers of message-source-nodes intuitively causes the high throughput, burdening the bus channel. The reason more messages being algebraically accumulated from these many bus coupled message-source-nodes is discussed in this section. Second, customized functional interrelations predefined inside the Application Layer also lead to the short-time high traffic state, which is discussed in Chapter 7.

Several distribution types of message-sending frequency, set as the model input parameter, has been sorted and fitted by goodness of fit method, mentioned in Section 2.3.4. Therefore, the expectation value of every two message-sending time interval in each message source is homogeneously set to 1.5 second. This parameterization

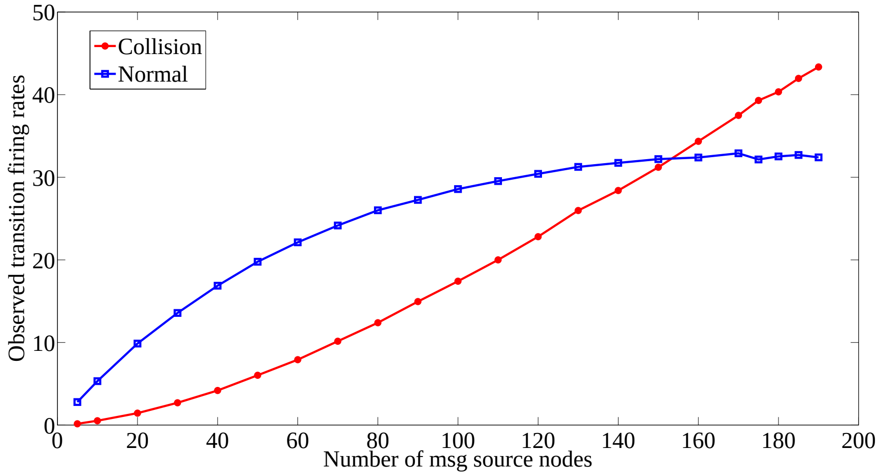


Figure 6.8: Firing rates of Bus normal behavior vs. Bus collision detection

approach is for the purpose of establishing a relatively constant and continuous system input. The transition rates are generated by Monte-Carlo simulations of various models with the message-source-node number ranging from 5 to 190.

6.3.1 Collision vs. Normal State

According to the token flow inside the communication model, all input messages pass through the model with two routines. They are collision transition and normal transition. Therefore, the result of model output is shown in Figure 6.8.

With the increasing message-source-nodes involved in one bus channel, the increasing number of both measured routine occurrences is inevitable, but one feature needs to mention that, the curve "*Normal behavior*" has a non-negative exponential distribution in the first phase and then approximates to asymptote of transition rate value 32. And the tendency of the collision rate increases with exponential growth. Another interpretation of this phenomena is shown in Figure 6.9, provided by firing rates of normal and collision behavior shown in Figure 6.8.

The ratio between the collision rate and the normal rate is a key parameter to availability analysis of busload validation. As we can see that, at the beginning, most messages occur on the bus channel with rare collisions; if the condition of total sending frequency is higher than 140 messages per second, the probabilities of these two transition firing rates are equal. According to the current SmallCAN specifications [Schrom et al., 2011], the maximum bus channel sending frequency is limited to 138 messages per second. This formal approach theoretically generates the

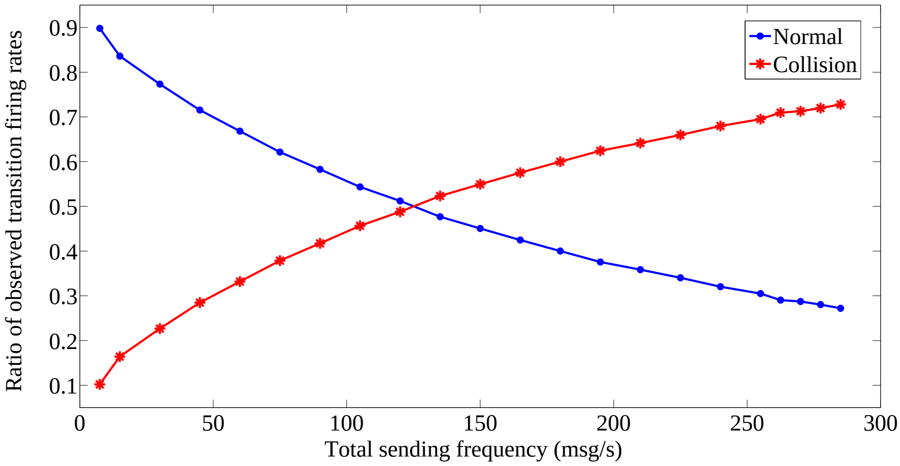


Figure 6.9: **Bus channel state vs. Total sending frequency**

worst-case scenario with conditions that sending frequency is even higher than the critical value of the real system, which is shown in the later phase of Figure 6.9.

6.3.2 Message Channel Occupation vs. System Input

Each message generated from the dedicated message-source-node requests to occupy the bus channel. But not all messages as bus system input may successfully get access to the bus channel under the circumstances of high traffic density. Therefore, the comparison of occupied messages between low and high traffic density is shown in Figure 6.10.

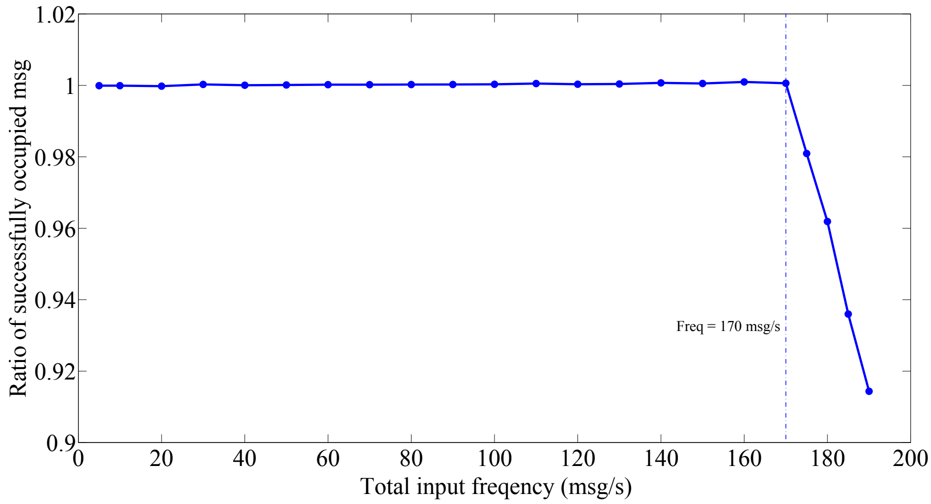


Figure 6.10: **Successfully occupied message vs. Total input frequency**

Due to the Figure 6.10, when the total frequency of system input is higher than 170 messages per second, the percentage of occupied messages compared to the total current input messages are sharply declined from 99.9% to 91%. According to the model structure, the decreasing phenomena does not mean that the utilization of the bus channel is degenerated. The reason for this is that the rest messages are sent to the bus channel via multiple access instead of waiting for single access routine. Therefore, the comparison between these two parameters is also necessary. This will be discussed within the next Section 6.3.3.

6.3.3 Single Access vs. Multiple Access

If two messages request to occupy the bus simultaneously within one frame-sending time interval, not only the collision state is triggered, but also the multiple access mechanism, designed and implemented in the model structure, is also triggered. This single / multiple access as part of the access mechanism can transfer messages in this phase immediately for further priority comparison without waiting. Current traffic of bus channel has a huge impact on the transition rates of these two parameters. In order to illustrate this phenomenon, two types of traffic as model input are simulated. The result is shown in Figure 6.11.

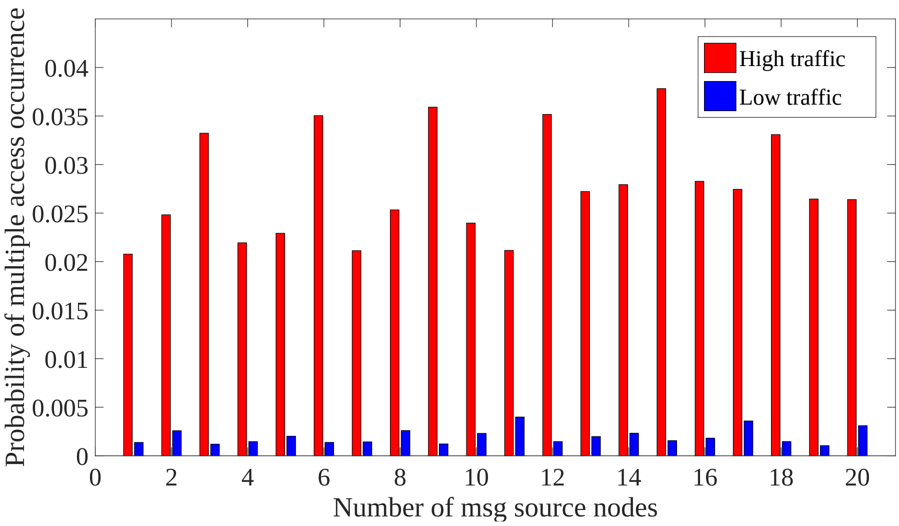


Figure 6.11: **Analysis of multiple access mechanism under high and low traffic density**

Due to CSMA mechanism, each node is listening to the bus state before sending messages on the bus channel. The percentage of multiple access shows a relatively low range between 0% and 0.5% under the low traffic scenario while under high traffic density scenario on the channel, the probability of triggering the multiple access transition is between 2% and 3.8%.

The reason for the apparently oscillating probability in Figure 6.11 is based on the concurrent sending characteristics of the nodes.

6.3.4 Priority Comparison vs. Message Source Numbers

In serial communication, the mechanism of message bitwise priority comparison is generally adopted: in the communication model, any concurrent messages-sending scenario is automatically enabling the priority comparison. It is important to analyze the result whether one relatively low priority message-source-node might be muted too long to send out its messages. It is also crucial to assign a priority to each message ID due to its functions in the design phase. The result of priority comparison from the 20 message-source-nodes Monte-Carlo simulation analysis is shown in Figure 6.12.

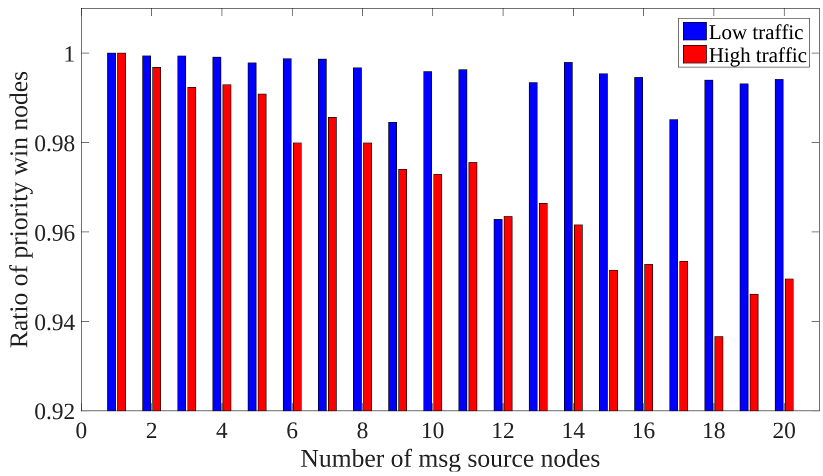


Figure 6.12: Performance of priority win under high and low traffic density

According to the model structure, message IDs are predefined by their message-source-nodes, the priorities of which have negative proportions to the values of message-source-node IDs. And it is clearly illustrated in Figure 6.12 that the probability of the message-source-nodes with low priority under high traffic circumstance can reach to 93.7%. On both conditions, messages with high priority arbitrarily win out, ensuring the messages with high priorities, such as system state checking messages or global command messages, successfully occupy the bus channel within the time constrains under the same range. It is worth mentioning that the curve "*High traffic*" decreases relatively more downwards than the curve "*Low traffic*", because of the input of the model is not deterministic, although the expectations of each sending frequency is fixed to the constant value.

Figure 6.13 shows the performance of priority comparison in a different dimension: the total message-sending frequency of accumulated the message-source-nodes

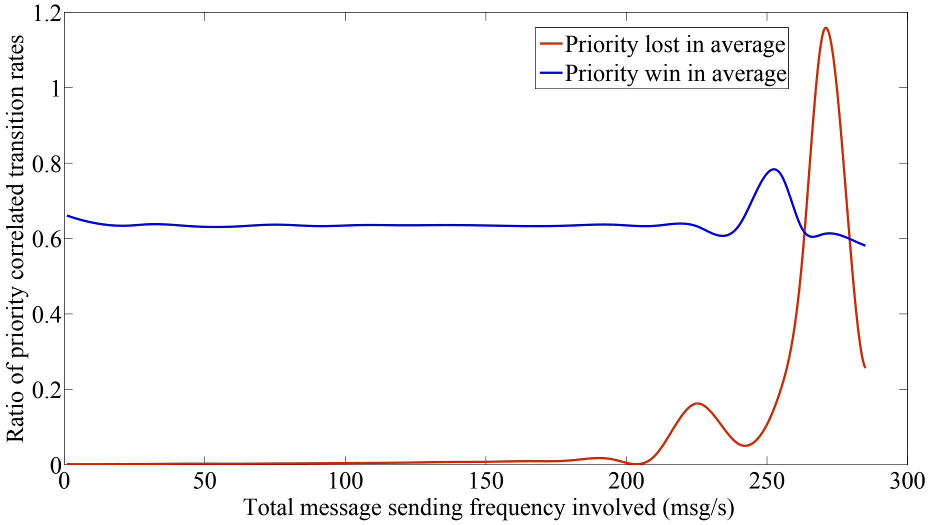


Figure 6.13: **Analysis of priority comparison mechanism in average**

is scaling with 0 $msg/s \leq Frq \leq 300\ msg/s$. The quantity "*Priority win in average*" is defined in Equation 6.10,

$$P_{\_win\_in\_average\_n} = \frac{\sum_{i=1}^{n} R_{\_Priority\_win\_node\_i}}{n} \tag{6.10}$$

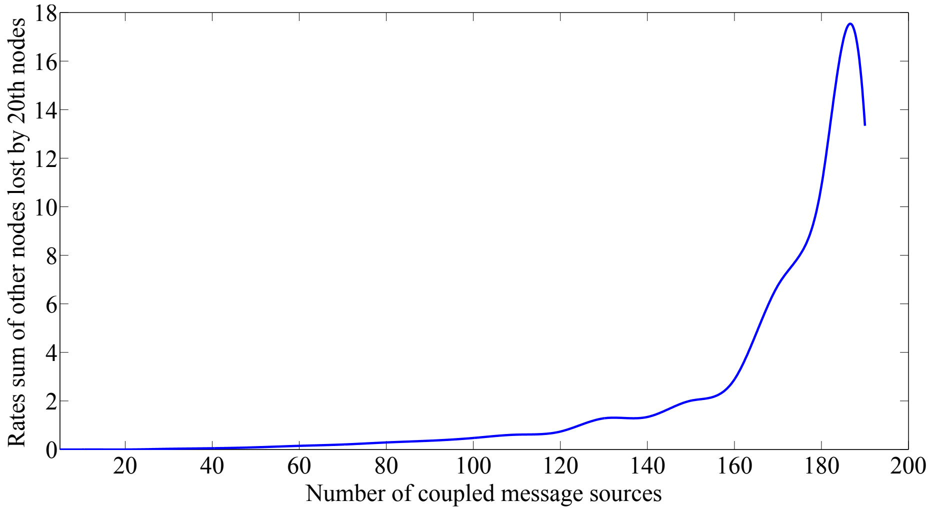
where n is the number of message-source-nodes involved in the communication model. The quantity "Priority lost in average' is calculated in Equation 6.11,

$$P_{\_lost\_in\_average\_n} = \frac{\sum_{i=j}^{n}\sum_{j=1}^{i} R_{\_Priority\_lost\_node\_j}}{\sum_{j=1}^{i} j} \tag{6.11}$$

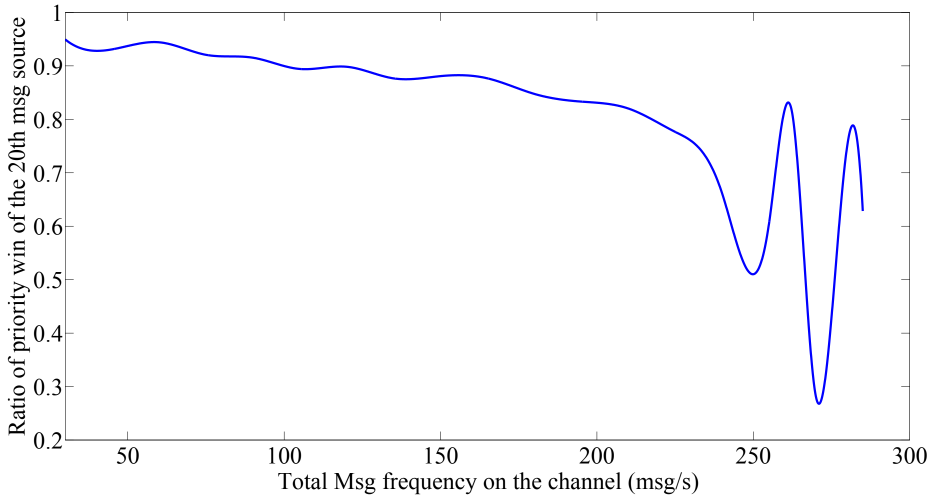
where $R_{\_Priority\_lost\_node\_j}$ represents the transition firing rate of any occurred priority lost event. For example, with 170 message-source-nodes involved in one fieldbus, the message-sending frequency has been accumulated to 255 msg/s and the total number of priority lost occurrences is up to 18383 times, whereas the total number of priority lost occurrences with the total message frequency 240 msg/s is up to 12719 times.

6.4 Performance Evaluation of the 20th Message-Source-Nodes

Performance evaluation approaches mentioned in last sections have carried out regarding the view of message throughput on the global bus channel. In this section, the performance also shed lights on the 20th message-source-nodes, characterizing out of the individual message-source-nodes.



(a) Priority lost of 20th message-source-nodes



(b) Priority win by 20th message-source-nodes

Figure 6.14: **Analysis of priority comparison mechanism of 20th message-source-nodes**

Compared with the discussion on the bus channel's priority comparison in Figure 6.12, the sub-figure a and b in Figure 6.14 show how the 20th message-source-nodes weights to influences on the global bus channel.

Moreover, in Figure 6.12a the X-axis is scaling up the message-source-nodes coupled into one fieldbus channel: 200 message-source-nodes are involved with the homogeneous message-sending frequency, similar parameter assignments mentioned in Section 6.2. The curve is subject to the exponential increase before reaching the higher frequencies. The fact that more proportion of other nodes failed in experiencing the priority comparison mechanism indicates that the 20th message-source-node becomes more important with more message-source-nodes connected to one bus channel. The hopping curve in the higher frequencies shows the backward tendency of its weight among message-source-nodes.

Figure 6.12b shows that under relatively low traffic density, messages from 20th message-source-nodes have the most successful channel occupation in average. It is worth mentioning that the transition rate of priority win in average is stabilize to the value of 0.65, compared with the priority win rate ranging from 0.95 to 0.85 under normal channel throughput, shown in Figure 6.14b. But system performance degrading occurs when the message-sending frequency exceeds 200 msg/s. However, the curve "*Priority win in average*" oscillates much less than the curve "*Priority lost in average*". It implies that the timed behavior related to priority win is further bottlenecked by the minimum time interval of transmitting on the bus channel. After the message wins out, it needs to be sent out in the final analysis. On the contrary, timed behavior related to priority lose is more free under current model construction. Therefore, it reacts in a more sensitive way with the increase of the total message-sending frequency.

6.5 Chapter Conclusion

The performability analysis regarding validation procedure in this chapter have been investigated and illustrated. The simulation results involving 200 message-source-nodes sub-models have been analyzed and evaluated on the aspect of the system performance. Results of this validation approach can be recorded as formally validated criteria in the fieldbus protocol for further system design.

Based on the performability assignments by the attribute hierarchy, the extended and parameterized Petri net communication model is simulated by the computerized simulation method. As a result, the transition rate analysis based on the Monte-Carlo simulation results has been carried out under relatively low and high traffic density scenarios. The busload of large scale fieldbus system has hereby been profiled and evaluated.

The characteristics influencing on the performance of large scale fieldbus system with both low and high traffic density have been quantified and analyzed respectively, such as message collision scenario caused by concurrent sending, efficiency of channel occupying, CSMA/CA and bitwise priority comparison. The evaluations of these quantities are based on the simulation results of varied traffic density of the extended Petri net communication model.

This evaluation approach not only validates the correctness of the model structure, but also provides quantified and validated criteria for further fieldbus protocol development. It worth mentioning the independence among different message-source-nodes makes it possible to implement the programming-based model extension. The function interrelation itself is analyzed and evaluated in Chapter 7.

Chapter 7

Observation and Analysis of the Emergent Message Burst

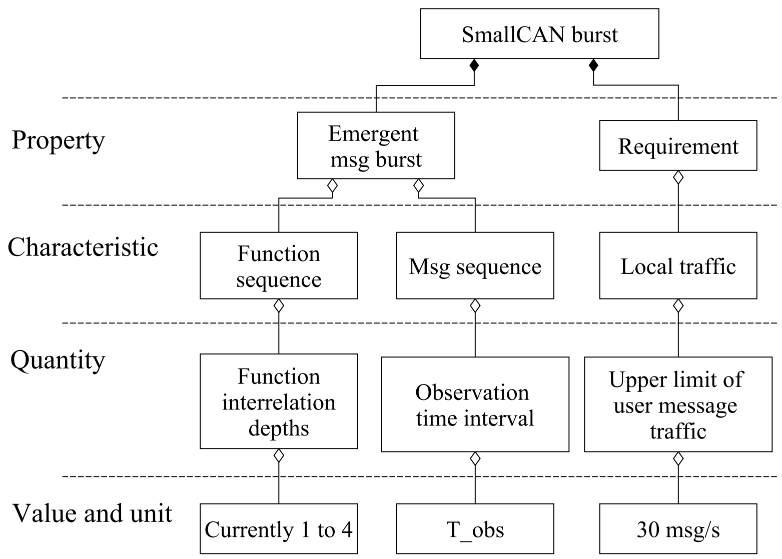


Figure 7.1: **SmallCAN burst assignments by the attribute hierarchy**

As shown in Figure 7.1, the SmallCAN specifies the local traffic from each networked message-source-node no more than $30msg/s$ [Schrom, 2003]. However, instead of analyzing the message-source-node, in this work the message-source-nodes are defined and categorized with the corresponding message IDs of executing the same function or function interrelations. Therefore, the focus of this chapter is on analyzing and evaluating the emergent bus behavior of large scale fieldbus system.

First, the formal description of functional interrelations, concerning the emergent property Definition 3.2.1 described in Section 3.2, is presented by Petri net's syntax and semantics. Second, a case study abstracted from the real fieldbus-based building automation system is then discussed to analyze the structure of functional interrelations, based on which the term function-interrelation-depth is hereby presented and discussed. It is further mapped to the extended Petri net communication model, generating the message sequence for the purposes of simulation and analysis. Finally, the observation approach is hereby discussed based on analyzing the sorted transition firing rates of the selected function-interrelation-depth in order to evaluate the functional complexity.

7.1 Formal Description of Functional Interrelation by Petri nets

With respect to the emergent property Definition 3.2.1 described in Section 3.2, the system relations can be formally presented by Definition 7.1.1.

Definition 7.1.1. System Relations [Mueller, J. R. and Schnieder, E., 2008]

In a system S with a set of objects $O = \{o_1, o_2, ..., o_n\}$, a SPN is defined as P, T, sT, P, F, m_0: two objects o_i and o_j, with $i \neq j$, interfere, if there is a possible state change from $s_{i_m}(o_j)$ to $s_{j_n}(o_j)$ such that the state change of $s_{i_m}(o_i)$ to $s_{i_n}(o_i)$ is a necessary condition.

The relation of o_i and o_j can be modeled in N by Equation 7.1:

$$\exists t \in T : \{p_{j_m}, p_{i_m}\} \subseteq {}^{\bullet}t \wedge \{p_{j_m}, p_{j_n}\} \subseteq t^{\bullet} \tag{7.1}$$

Semantic description of emergent property using Petri net theory is correspondingly presented here:

Let $PN_1 = (P_1, T_1, sT_1, F_1, \Pi_1, m_1)$ and $PN_2 = (P_2, T_2, sT_2, F_2, \Pi_2, m_2)$ be two stochastic Petri nets, with $P_1 \cap P_2 = \emptyset$ and $T_1 \cap T_2 = \emptyset$.

Let PN_{12} be defined as $PN_{12} := (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, \Pi_1 \cup \Pi_2, (m_1, m_2))$ and let further be PN_3 be defined as $PN_3 := (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, \Pi_1 \cup \Pi_2, (m_1, m_2))$ with $F_1 \cup F_2 \cup (P_1 \times T_2) \cup (P_2 \times T_1) \cup (T_1 \times P_2) \cup (T_2 \times P_1) \subseteq F_3 \subseteq F_1 \cup F_2$.

Hence $[(m_1, m_2) >_{PN_3} \subseteq [(m_1, m_2) >_{PN_{12}}$, *i.e.* in comparison to the behavior of PN_{12} via $[(m_1, m_2) >_{PN_{12}}$, the behavior of PN_3 has been generally restricted at least, defined by the relation $(P_1 \times T_2) \cup (P_2 \times T_1) \cup (T_1 \times P_2) \cup (T_2 \times P_1)$. If this restriction realizes a new function on the PN_3 system level, then this new function behaves as an emergent property of PN_3.

7.2 Structural and Temporal Analysis of Functional Interrelations

According to the work of [Kiefer, 1996], the exact relationship between the abstract and real system being used to describe the relations termini allocation and partitioning is defined as follows:

Allocation: resources assigning to the system functions before commissioning or during operation, for example, transmitting message (*i.e.* system function) and fieldbus nodes (*i.e.* system resource).

Partitioning: assigning a function or group of functions to system in several units / modules, geometric / structural collocation or system performance. For example, executing a control task (*i.e.* system function) while the interaction of technical information system and operator (*i.e.* system resource).

One proper case study concerning functional allocation and partitioning is the one application module as one message-source-nodes named "Energy and Power Evaluator" designed and implemented on the real fieldbus-based building automation system SmallCAN. Based on this, several research approaches concerning measuring and predicting the energy consumption have been illustrated, as mentioned in the work of [Diekhake et al., 2012] [Kurczveil et al., 2012] and [Kurczveil et al., 2014].

One of its functions regarding this section is to record and update the values of the energy consumption and the current power from the entire system as well as from the networking fieldbus components. The volume of energy value in this case is designed in capable of holding its long-term value. It consists of six bytes data for one time energy-updating procedure. According to the SmallCAN frame definition in Section 4.3, the data field is only 2 bytes with the extra data type definition, moreover, this 2 bytes data field still contains the data types, such as integer and double float. As a result, this long data is then byte-wise divided into six frames. Therefore, its energy recoding and updating function is proceeded by functional interrelation, generating a continuous message sequence combining these six messages onto the bus channel.

The scenarios of triggering the short message peak are defined as follows:

- protecting data in case of system emergency, such as system recommission after irregular shut down
- regularly updating energy consumption value
- reading in the energy value saved in the EEPROM (Electrically Erasable Programmable Read-Only Memory) of the relevant bus coupler

The predefined functional interrelations trigger the message sequencing and thus cause a short-time message burst on the bus channel whenever the system states matches the scenario mentioned above. These abnormal occurrences, such as the message-sending sequences, are hence necessary to be analyzed to limit them under certain quantities.

An important term of hierarchically classifying the functional interrelations need to be mentioned is hereby defined as function-interrelation-depth. The function-interrelation-depth concept is derived from analyzing the log data from the real fieldbus-based building automation system. The term is defined for describing the emergent bus behavior of fieldbus system.

As is shown in Figure 7.2, a case study of the selected function-interrelation-depths in real fieldbus are introduced, containing HVAC functions (Heating, Ventilating, and Air conditioning).

Temperature control combined with temperature sensors and motion detector consists of the first function-interrelation-depth. Intuitively, the first depth receives information and influence from the system context. Once the first function-interrelation-depth is triggered, executing the following function-interrelation-depths is inevitable. Therefore, this layer also plays the role of function supervision. According to the work of [Diekhake and Schnieder, 2013], it provides an on-line supervision possibility of connecting the real system and its formal model.

The condition of transforming the function-interrelation-depth 1st and 2nd is the current temperature varies across the confidence of the defined criteria based on the condition of activities triggering the motion detector. The 2nd function-interrelation-depth is mainly composed of implemented Free-locating Special Functions (FSFs). As is mentioned in Section 3.1.1, FSF is designed as a functional interrelation realization implemented in fieldbus-based building automation systems. FSF is released from designate locating hardware, and thus can be burned into any fieldbus coupler. It receives the function-control messages from the pre-function relevant message-source-nodes. Then With the help of its inner logic, FSF calculates, configures and sends out messages to the further function relevant message-source-nodes.

The selected function mappings are shown in Figure 7.2 drawn by UMLCD, the function depths can be basically categorized into three main functional interrelations: supervision functional interrelation, FSF interrelation and execution functional interrelation. The supervision functional interrelation is composed of function comparisons whether to trigger the next level, although some sensors and actuators are also involved in this level, but the purpose is still to trigger the FSF functional interrelation if meeting the criteria provided by those endpoint devices. And then FSF functional interrelations nodes configure the following function sequence routine based on the information provided from the supervision functional interrelation

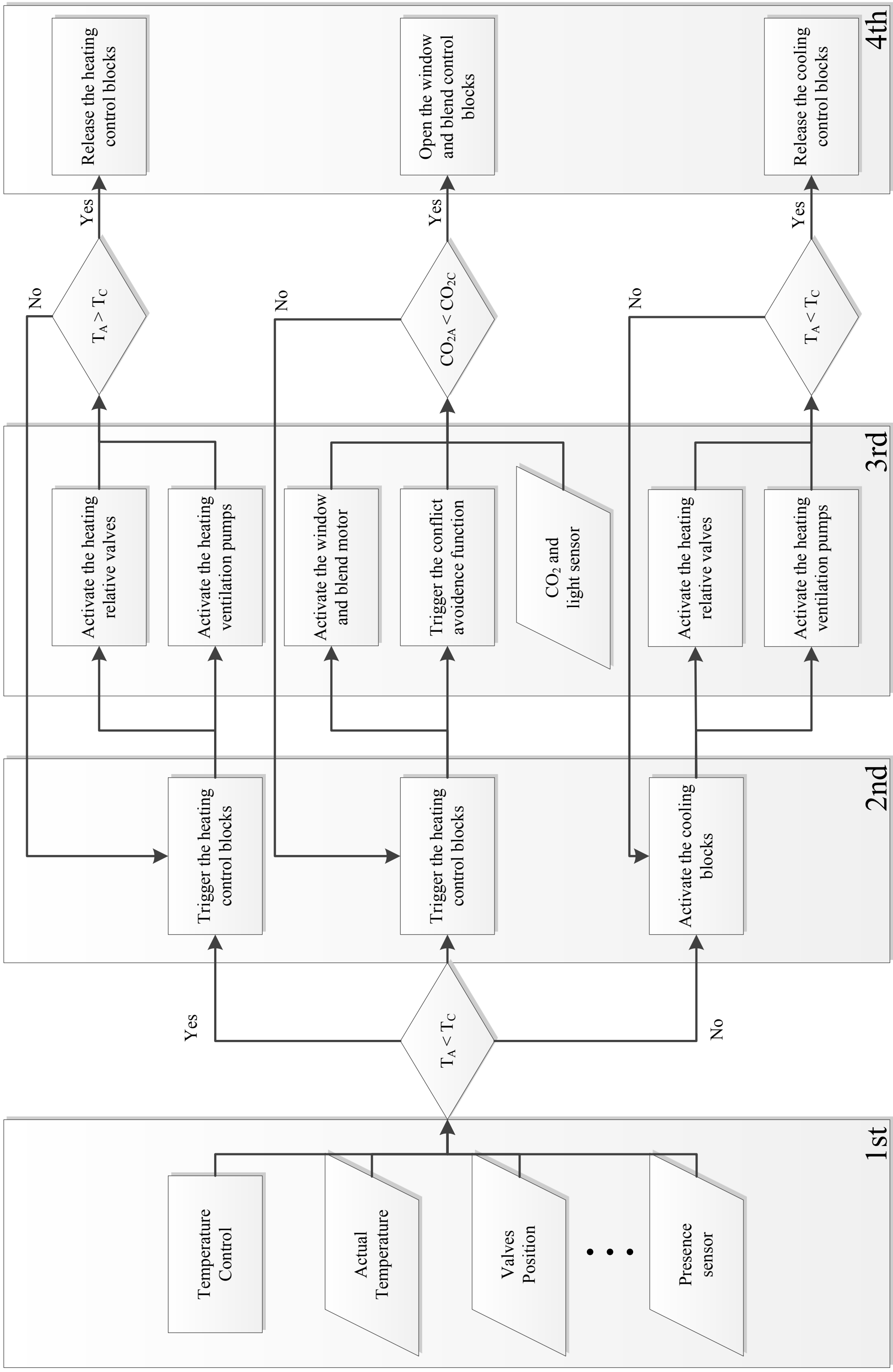


Figure 7.2: **FLow chart of HVAC control categorized by function-interrelation-depths**

nodes, which will finally be executed by the drain devices of third functional interrelation nodes. For example, the comparison of the current temperature provided by message from the message-source-node P5 leads the function sequence to pass through the heat function or cooling down. The prerequisite is still that the moving sensor message from message-source-node P11 is triggered by someone in the house. All Valves messages related to heating and cooling process from either message-source-node 14 or message-source-node 17 support the function assignment in the second function-interrelation-depth.

In this case study, the FSF functional interrelation mentioned in 7.3 gathers the supervision messages as input sources of making decisions, deduces with its inner-logic and sends the messages to 3rd function-interrelation-depth as commanding the drain devices, or allowing to enter further function execution, such as message-source-nodes in charge of heating and cooling functions, including pumps and ventilations as well as message-source-nodes for CO2 emissions and lightness sub-functions. In the end, 4th function-interrelation-depth is also composed of message-source-nodes with FSF functions relevant. The criteria whether this depth transforms to further interface or recycle to former depth are also depend on mainly on temperature and air quality inside the office room.

This structure is abstracted only considering the temperature effects, controlled by predefined functional bindings inside the Application Layer, due to the functional requirements in different scenarios. These function bindings are executed with orderly generated message sequences, which cause a short time burst on the fieldbus channel. This may lead to the system performance degradation. Therefore, it is necessary to quantify and validate the key functional interrelations, ensuring the predefined. The four layers of functional interrelations are hereby named as functional depths, semantically interpreted as $\{f_{d1}, f_{d2}, f_{d3}, f_{d4}\}$.

These functional interrelations already exist inside the fieldbus system configuration and are performed by generating a continuous messages' sequence. So the function-interrelation-depths of the system inside the Application Layer can effect a the message burst on the bus channel. The functions of each block in Figure 7.2 are composed of one or more messages triggered by the function-interrelation-depth in order. If one message occurs inside one function, then it triggers other relevant functions according their sporadic interrelation-depth's position.

It can be concluded from this case study that two types of fieldbus messages mainly composed of these function-interrelation-depths: messages in charge of function supervision and function execution. Therefore, further integration and analysis of extended Petri net communication model combined with these two kinds of function-interrelation-depths are of great importance for evaluating the corresponding type of busload. More discussions can be found in Section 7.3 and Section 7.4.

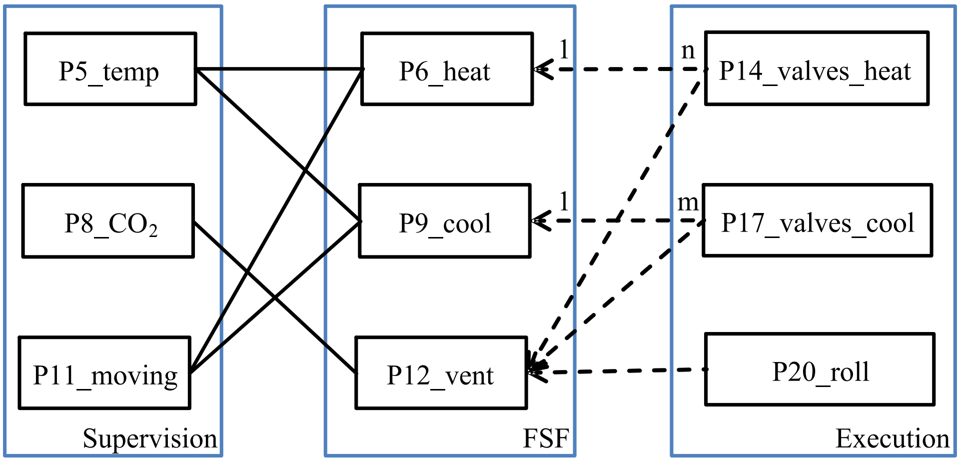


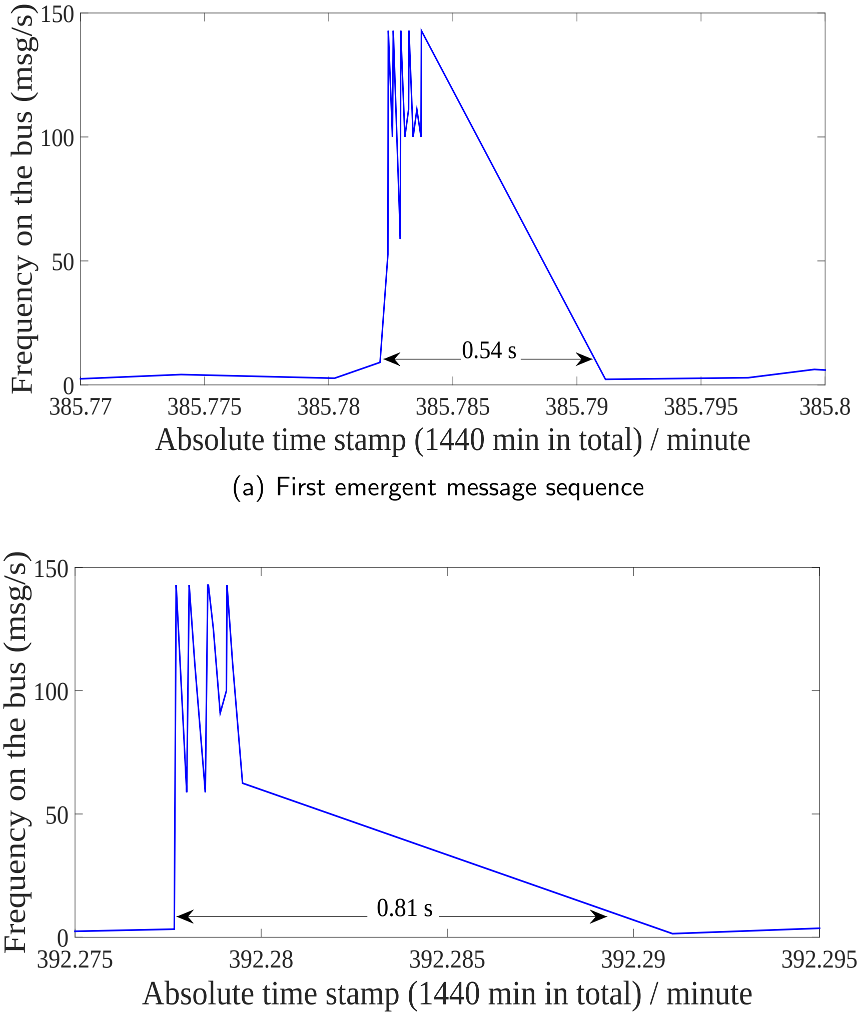
Figure 7.3: **Classfication of the 1st function-interrelation-depth by UMLCD**

The case study of function-interrelation-depths is generated from the log data. Each message sequence begins with a specific message ID. Once it occurs on the bus channel, the message sequence will occur afterwards. Therefore, the message sequence are measurable by detecting the occurrences of this specific message ID.

As is shown in Figure 7.4: it shows the message sequence abstracted from the function executing messages observed on the bus channel. This message sequence is designed for controlling the temperature. It has been predefined and implemented inside the Application Layer. It is one of the paths described in Figure 7.2.

The emergent message burst, according to the emergent property Definition 3.2.1 described in Section 3.2 and the formal system relations' Definition 7.1.1, is generated from observing the focused function-based message sequence in log data. It is quantified as the time interval of one function-related messages sequence successfully transmitting from the source and to the sink, which fulfills the defined functions inside the fieldbus APL. By observing the focused message sequence, the emergent burst event in log data starts with the sending event of message ID 15500, followed by executing a serious short-term message-sending events.

As shown in Figure 7.4, the time intervals of these two message sequences are 0.54 *second* and 0.81 *second*. The time difference indicates that the function-execution paths of these two message bursts are not identical. This is caused by entering and executing the different functional interrelations depths although the bust events are triggered by the same function-starting message ID. To conclude, various environmental conditions generate different message sequences.



(a) First emergent message sequence

(b) Second emergent message sequence

Figure 7.4: **Emergent message sequences sorted from SmallCAN's log data**

7.3 Model Integration of Variable Function Relation Depth by Petri nets

Function supervision messages as a part of the application system message input has been built based on the extended communication model by π-Tool. The message-sending behavior of the source nodes grouped with the same function sequence has been bound within one abstracted Petri net structure. The message sending frequencies of the bounded source nodes are determined and synchronized by the input transition "FSF trigger". Figure 7.5 shows the abstracted Petri net structure of function-interrelation-depth.

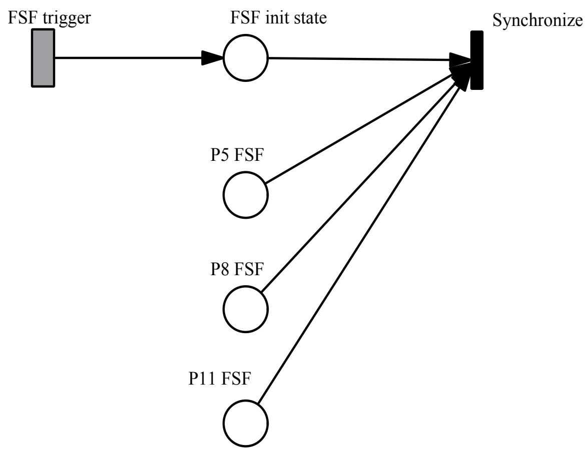


Figure 7.5: **Petri net model input with 1st function-interrelation-depth**

Places “P5 FSF”, “P8 FSF” and “P11 FSF” are the synchronizing places fused with the post set of first input transition of each FSF node, shown in Figure 7.5. All the FSF related message-source-nodes have the same access mechanism structure and share the same bus channel state compared with independent node in common. The focus in Figure 7.5 is to reveal the FSF message trigger structure by fusion places such as place “Fusion FSF init state” and places “Fusion P5 FSF”, ”Fusion P8 FSF” and ”Fusion P11 FSF”. When all relevant messages according to the function sequence are triggered for further message access, the transition “Synchronize” consumes all tokens from pre-places of this FSF structure and releases the FSF message trigger process for the next step.

The FSF structure is generalized by main function chains controlled by the number of message-source-node. Their own interrelations result in a burst of message sequences, which are detected and sorted from the log data of a real SmallCAN field-bus system application, implemented in one office named ”Future Workspace”, see the work of [Diekhake, P., Liu, J., and Schnieder, E., 2011].

As modeled shown in Figure 7.5, three message-source-nodes are bound with a special function group with node ID 5, 8 and 11. Therefore, transition “FSF trigger” plays the role of system input and set a token to the place “FSF init state”. This place is fused with input places of the message-source-nodes, which belong to this special function sequence. In this case, these nodes ID are also 5, 8 and 11. In other words, place “FSF init state” is the input of all three message-source-node sub-models.

The time interval of triggering this FSF event is determined by the time parameterization of transition “FSF trigger”. This is sorted from the log data observing the time interval of the focused message sequence predefined inside the Application Layer. Compared with the independent message-sending nodes, each time interval defined in the initial transition of each independent message-source-node has its own

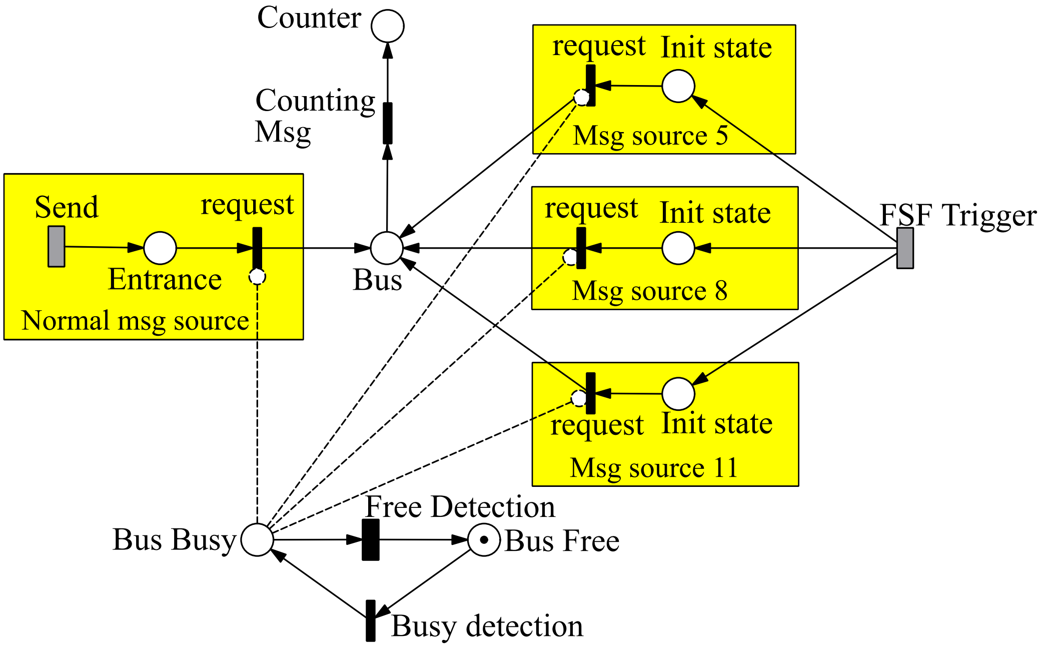


Figure 7.6: Petri net structure of the 1st function-interrelation-depth

distribution type with expectations and variances. This is defined inside whereas each FSF message sequence is triggered by only its initial event, which results in the only time concept of all FSF relevant message-source-nodes.

As is mentioned above, the message sequence is performed by messages passing through the function-interrelation-depth in right order. And once this routine is triggered, the following step necessarily occurs with its order. The extended communication model with function supervision messages is further extended with typical messages in charge of its function executions involved.

Figure 7.7 extends the triggering message-sending scenario of Figure 7.5 with the 2nd and 3rd function-interrelation-depths. The token in place "FSF init state" firstly activates the 1st function-interrelation-depth nodes with ID 5, 8 and 11, the three places "P5_temp_1", "P8_compare_1" and"P11_moving_1" are filled with feedback tokens after the messages in these nodes are successfully activated. *I.e.* the scenario of completing the first function-interrelation-depth is mapped by these four placed mentioned above with tokens set inside each place of the Petri net model in Figure 7.7. The following function-interrelation-depths will be automatically triggered respectively by the related nodes in former depth. In this case the 2nd function-interrelation-depth will be finished if the mapped fusion places with name "P6_heat_2","P9_cool_2" and "P12_vent_2" are set with tokens. The last step in this model is the same procedure mentioned above. The places "P14_roll_3", "P17_vavles_3" and "P20_co2_3" mapped with specified messages for function execution are set to tokens, because they are triggered by the function supervision messages

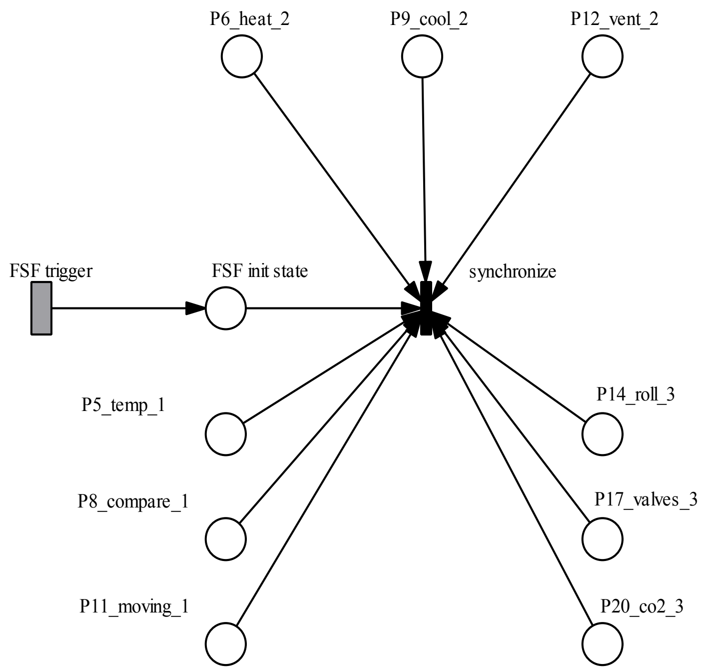


Figure 7.7: Petri net model input with 1st, 2nd and 3rd function-interrelation-depths

generated from the 2nd function-interrelation-depth to execute the functions in the drain device of the field bus system.

It is worth mentioning that all these nine places fused with message-source-node IDs orientate not for message-sending procedure but the feedbacks after messages are triggered in all related nodes. Once all places in Figure 7.7 are set, it will immediately be consumed by the transition "Synchronize" to end this message triggering structure. Only one time parameter exists here, that is the mean time interval between every two function sequences.

7.4 Evaluation of the Marked Token Flow

Further evaluations of the function-interrelation-depths' structure and the extended communication model have been constructed by Petri net. The attempt is to observe the focused transition firing path described by the model structure in Figure 7.7. It is performed by related message-sending sequences until the last related message winning out of the current messages during the priority comparison process.

Two observing approaches, involving only the 1st function interrelation depth and involving the 1st, 2nd and 3rd function interrelation depths, are hereby discussed. By observing the marked token flow related to these two function interrelation depths involved, their time intervals between every two focused events during the model animation are recorded and sorted respectively based on a large number of observation approaches. The purpose of observing time span is to provide validated criteria to improve the fieldbus protocol for function design and limiting the number of further functional interrelations.

These two function-interrelation-depths triggering structures bound with the extended communication model are observed respectively. This is realized by observing the focused marked token flow in the token animation function integrated inside π-Tool. 100 observation tests with the busload model combined with 1st function-interrelation-depth have been collectively completed, the result shows in Figure 7.8.

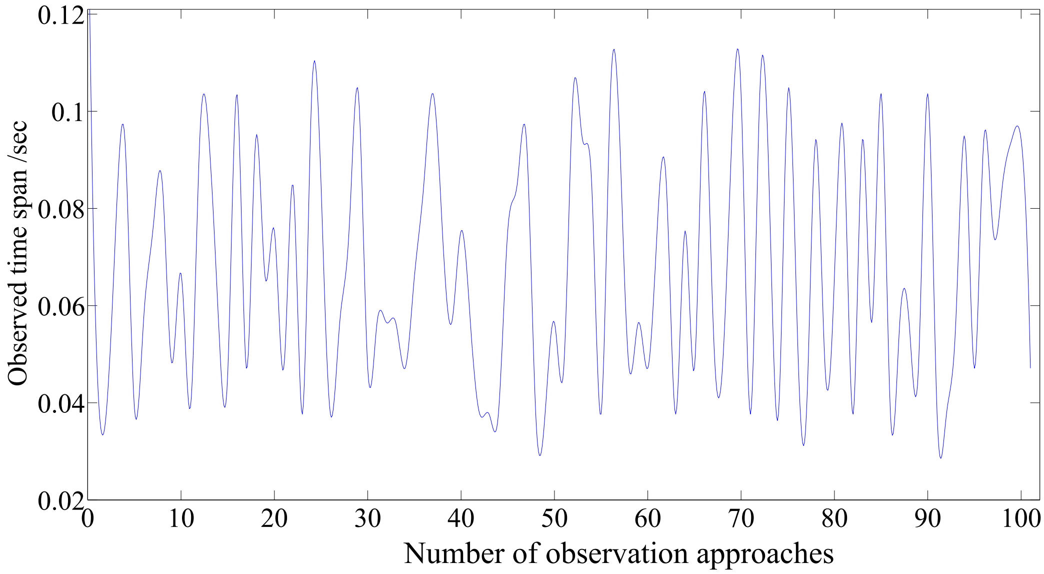


Figure 7.8: Observation result of the 1st function-interrelation-depth

The time span was recorded between the time stamp of firing transition "FSF trigger" and the time stamp of firing transition "Priority win" of the last function-interrelation-depth relevant message from message-source-node ID 11 as one observation approach point shown in Figure 7.8. Such observation started from manually firing transition "FSF trigger", then messages from the triggered model were assigned to three related message-source-nodes, the current messages on the bus channel combined with generated messages began to pass through the extended communication model for occupation and further access. The observation approach stopped at the time stamp when the last message related to trigger model from message-source-node ID 11 completed its access process. The observed time span is calculated between the two recorded time stamps. The random variance among the observation path is due to the current uncertain traffic composed by the number of bus channel occupying messages with stochastic behavior of each independent sending nodes involved. In addition, the priorities assigned by these messages are also not fixed when transition "FSF trigger" is firing.

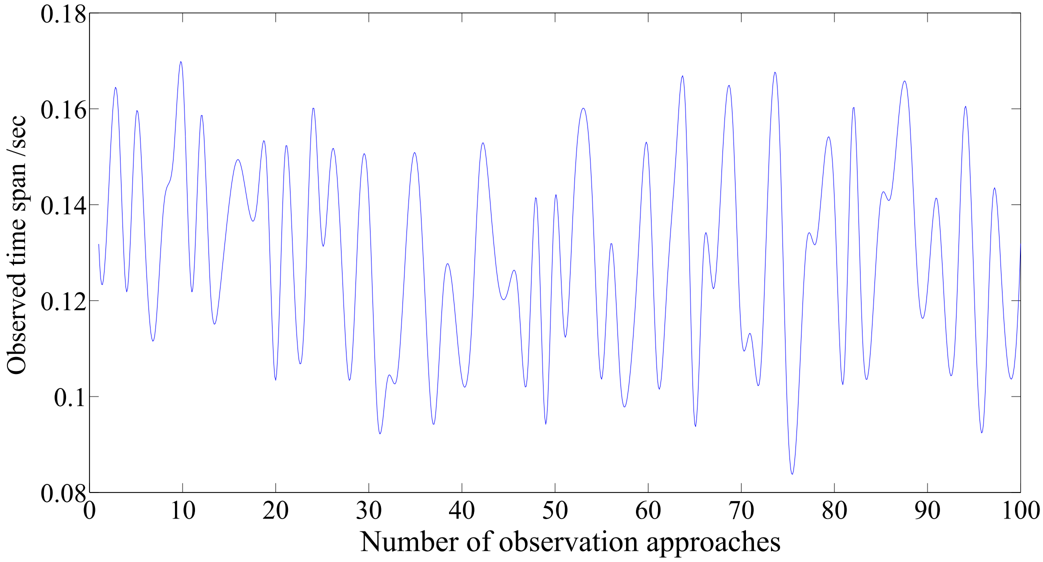


Figure 7.9: Observation result of the 1st, 2nd and 3rd function-interrelation-depths

The complete function-interrelation-depths are also observed with the same method mentioned above. As is shown in Table 7.1, the time span scale of 100 times observations is between 0.0942 second and 0.1678 second in the complete function-interrelation-depths, whereas the same parameter is between 0.039 second and 0.1034 second. For each observation approach, the mean time interval of every two priority winning messages is also recorded as 0.00942 second among all the observed firing transitions. It means that the current bus channel needs to schedule minimum 4 messages to maximum 11 messages to pass through the 1st function-interrelation-depth, while it needs to schedule minimum 10 messages to maximum 14 messages to complete all three function-interrelation-depths.

Table 7.1: Performance parameters related to the emergent bus behavior

Quantities influencing the performability	1st depth /s	1st, 2nd and 3rd depths /s
Minimum time span	0.039	0.0942
Maximum time span	0.1034	0.1678
Time span necessary	0.0283	0.0848
Minimum messages as bus traffic	4	10
Maximum messages as bus traffic	11	14
Necessary messages to schedule	3	9

7.5 Chapter Conclusion

This chapter focuses on the emergent message burst caused by functional relations. System relations regarding the emergent property Definition 3.2.1 in Section 3.2 have been formally defined. Then, a case study focusing on the emergent message burst has been proposed, functional relations of which are abstracted from a real SmallCAN system in building automation. The quantities of generated function-based message sequence has been measured and discussed, such as time intervals and typical function paths predefined inside the APL.

Furthermore, these function paths in the case study have been hierarchically structured using the defined term function-interrelation-depth. With different function-interrelation-depths, the interactions and function relations among involved message-source-nodes are discussed by UMLCD and later integrated to the extended Petri net communication model.

The token animation function inside π-Tool is hereby applied to generate the dynamic behavior of the Petri net model. By observing the focused transition firing sequences of two function interrelation depths involved, the time intervals of the these focused token flow have been observed and profiled.

With the aspect of emergent message burst, this observation approach can be applied as an upper criterion of defining the maximum amount of functional relations as well as function interrelation depths inside the Application Layer. As a result, emergent message bursts are restrained to occupy on the bus channel.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

This work provides a complete busload validation procedure with the aspects of performability in large scale fieldbus system. The conceptualization process of the validation procedure is outlined in Figure 8.1.

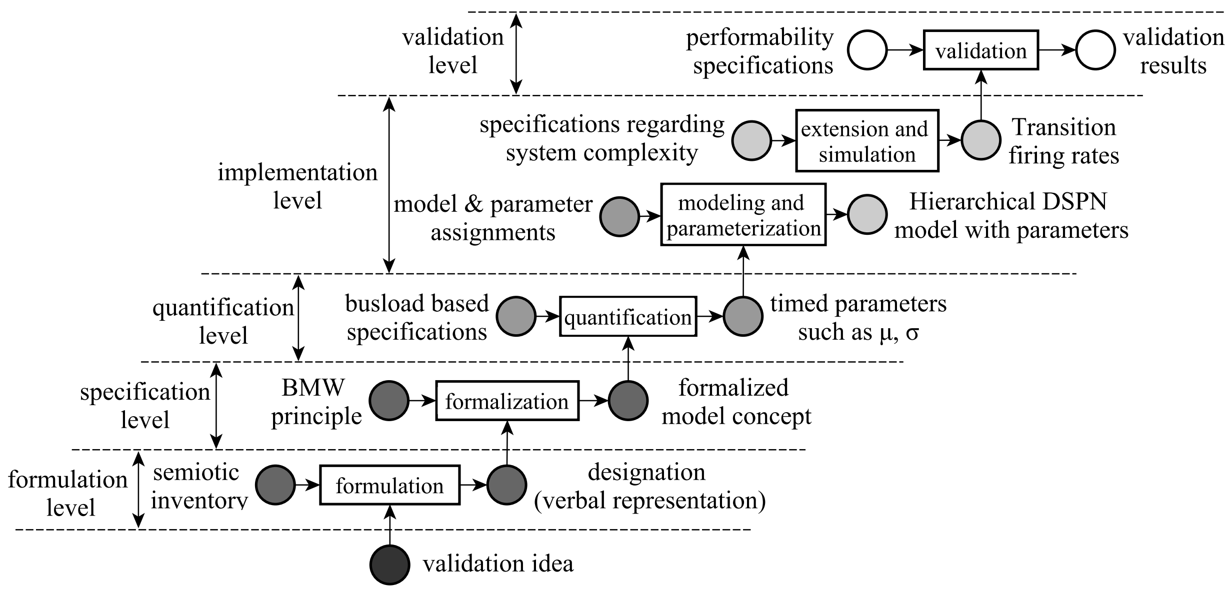


Figure 8.1: Process of conceptualizing the busload validation procedure based on [Schnieder et al., 2009]

In order to present the performability in a large scale fieldbus system, the validation procedure starts with an semantic idea in the formulation level. It is then designated into general system requirements by verbal representations.

Formalization is applied in avoiding conceptual ambiguities caused by diverse interpretations generated from the perception and specifying the validation procedure in detail. Therefore, suitable description means and methods are presented in the context of a fieldbus-based building automation system named SmallCAN. Attribute hierarchy theory incorporated with focused subsystem layers in OSI Model has been applied in formulating the SmallCAN specifications related to this work. The BMW principle together with system theory is selected in order to gradually structure busload validation procedure in a formal manner. Outlining the validation steps pave the way for analyzing the concurrent message-sending scenario.

Quantification of SmallCAN requirement regarding its busload is specified with the attribute hierarchy. It is then presented as follows. First, the dynamic property between the SmallCAN system and its context has been profiled. Based on this, quantitative analysis has been carried out for characterizing the busload behavior. Second, the selected message access mechanisms, such as CSMA/CA and bitwise arbitration, have been quantitatively analyzed under concurrent-sending scenarios. Third, focusing on profiling the concurrency, message-sending behavior have been abstracted and characterized by sorting the log data from the SmallCAN server implemented in real building office. These stochastic characteristics have also been further quantified into PDFs. Moreover, the goodness of fit method has been applied in fitting the stochastic distribution types among these PDFs, the expectations and variances of which are provided by the highest likelihood values. Finally, channel concurrency has been mathematically defined and its occurring probabilities are hereby calculated based on the fitted distribution types.

Regarding model-based implementation, the net structure and modeling assignments have been outlined. In addition, the Petri net modeling environment named Π-Tool together with its analytical characteristics is introduced. The concurrent sending-scenario as well as message access mechanisms, based on the quantitative analysis, have thus been modeled in a hierarchical way. Two kinds of Petri net subnet models, channel state sub-model and message-source-nodes sub-model, have been correspondingly constructed. Necessary net elements and their interactions involved in constructing the communication model have been discussed. Furthermore, these two kinds of Petri net sub-models together with their interactions have been synthesized into one hierarchical communication model. This model is named as the Petri net communication model, from which the complexity of the large scale fieldbus system can thus be characterized as: message-sending concurrency and emergent message burst.

Analysis of message-sending concurrency is based on the approach of the flexible model extension, where up to a maximum 200 message-source-nodes of Petri net sub-models and their interactions are further integrated with the channel state Petri net sub-model, providing variable traffic density. Based on the performability assignments by the attribute hierarchy, the extended and parameterized Petri net communication model is simulated by the computerized simulation method. As a result, the transition firing rate analysis of the focused characteristics, such as real-time analysis of concurrent sending, efficiencies of channel occupying, CSMA/CA and bitwise priority comparison, has been carried out under relatively low and high traffic density. The busload of large scale fieldbus system has hereby been evaluated with the aspect of performability.

Emergent message burst has been formally defined. In order to describe this emergent system property in detail, a case study abstracted from the real Small-CAN system in building office has been presented. Moreover, the term of function-interrelation-depths has been defined for structuring and quantifying the emergent bus behavior. By integrating different structures of function-interrelation-depths into the extended Petri net communication model, the observation times of selected token paths have been depicted and evaluated.

For further fieldbus protocol development and system validation, results generated from this validation procedure can be seen as the quantified and validated criteria to be recorded in fieldbus protocols for profiling and constraining the system complexity and ensuring system performability.

8.2 Outlook

The formal busload validation procedure provided in this work is adaptable for other applications.

Concerning the further quantitative analysis of message-sending concurrency, various stochastic message-source-nodes need to be involved, providing one large fieldbus system with more hybrid deterministic and stochastic message-sending characteristics. Meanwhile, the computerized resource need to be equivalent to execute the simulation. Therefore, the programming-based adjustable model extension could be optimized.

Moreover, further analysis regarding emergent bus behavior, the numbers of dynamic function-interrelation-depths, could be predefined and implemented in building automation systems with more complex function interrelation depths. Based on this, log data-based off-line analysis can be further proceeded, which in turn validates the on-line supervision result provided by [Diekhake, 2016].

Bibliography

[Ajmone Marsan et al., 1984] Ajmone Marsan, M., Conte, G., and Balbo, G. (1984). A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems (TOCS)*, 2(2):93–122.

[ASHRAE, 2005] ASHRAE, A. S. (2005). Standard 135-2004: BACnet - a data communication protocol for building automation and control networks. *American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA.*

[Bobbio et al., 2008] Bobbio, A., Gribaudo, M., and Telek, M. (2008). Analysis of large scale interacting systems by mean field method. In *Quantitative Evaluation of Systems, 2008. QEST'08. Fifth International Conference on*, pages 215–224. IEEE.

[Booch et al., 1999] Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The unified modeling language user guide*. Addison-Wesley, Reading and Mass.

[Bosch, 1991] Bosch, R. (1991). Can Specification Version 2.0. *Rober Bousch GmbH, Postfach*, 300240.

[Burton et al., 1988] Burton, P. et al. (1988). Field bus based on MIL-STD-1553B: Proposal to isa-sp-50. *ERA Technology Ltd.(Apr. 6, 1988) ISA/SP50-1988-148*, pages 1–125.

[Carl Adam Petri, 1962] Carl Adam Petri (1962). *Kommunikation mit Automaten.* PhD thesis, Technische Universität Darmstadt, Hamburg, Germany.

[Chen et al., 2012] Chen, L., Tao, T., Xianqiong, Z., and Schnieder, E. (2012). Verification of the safety communication protocol in train control system using colored petri net. *Reliability Engineering & System Safety*, 100:8–18.

[Chouikha and Schnieder, 1998] Chouikha, M. and Schnieder, E. (1998). Modelling of continuous-discrete systems with hybrid petri nets. In *Proceedings of the IFAC Congress*, volume 99. Citeseer.

[Dias Pereira, 2004] Dias Pereira, J. M. (2004). A fieldbus prototype for educational purposes. *Instrumentation & Measurement Magazine, IEEE*, 7(1):24–31.

[Diekhake, 2016] Diekhake, P. (2016). *Systematische Modellierung und Analyse eines verteilten Automatisierungssystems*. PhD thesis, Technische Universität Braunschweig, Braunschweig and Germany.

[Diekhake et al., 2012] Diekhake, P., Kurczveil, T., Liu, J., and Schnieder, E. (2012). Energieverbrauchsmessung haustechnischer komponenten mittels feldbussen der gebäudeautomatisierung. *Tagungsband*, pages 674–682.

[Diekhake and Schnieder, 2013] Diekhake, P. and Schnieder, E. (2013). Online monitoring of a distributed building automation system to verify large sequences of bus messages by causal petri net models. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 3651–3655. IEEE.

[Diekhake and Schnieder, 2015] Diekhake, P. and Schnieder, E. (2015). Strukturierte modellierung, simulation und überwachung verteilter automatisierungssysteme. *at-Automatisierungstechnik*, 63(2):129–142.

[Diekhake, P., Liu, J., and Schnieder, E., 2011] Diekhake, P., Liu, J., and Schnieder, E. (Magdeburg, Germany, Sep. 2011). Buslast- und schaltungssimulation zur validierung des optimierte feldbussystems smallcan bei maximaler auslastung. *KOMMA 2011 - Kommunikation in der Automation*.

[DIN61165, 2007] DIN61165 (2007). Anwendung des Markoff-Verfahrens, Beuth Verlag GmbH. *Berlin, e. V. DIN EN 61165, 2007-02*.

[Ehrig, 2003] Ehrig, H. (2003). *Petri net technology for communication-based systems: advances in Petri nets*, volume 2472. Springer Science & Business Media.

[EN50090, 2005] EN50090 (2005). Home and building and building electronic system (HBES). *CENELEC*.

[EN50129, 2003] EN50129, C. (2003). Railway applications-communication, signalling and processing systems-safety related electronic systems for signalling. *British Standards Institution, United Kingdom. ISBN*, pages 0580–4181.

[Fay and Schnieder, 1999] Fay, A. and Schnieder, E. (1999). Knowledge-based decision support system for real-time train traffic control. In *Computer-Aided Transit Scheduling*, pages 347–370. Springer.

[Felser, 2005] Felser, M. (2005). Real-time ethernet–industry prospective. *Proceedings of the IEEE*, 93(6):1118–1129.

[Flammini et al., 2002] Flammini, A., Ferrari, P., Sisinni, E., Marioli, D., and Taroni, A. (2002). Sensor interfaces: from field-bus to ethernet and internet. *Sensors and Actuators A: Physical*, 101(1):194–202.

[German, 2000] German, R. (2000). *Performance analysis of communication systems with non-Markovian stochastic Petri nets*. John Wiley & Sons, Inc.

[Gilks, 2005] Gilks, W. R. (2005). *Markov chain monte carlo*. Wiley Online Library.

[Güngör et al., 2011] Güngör, V. C., Sahin, D., Kocak, T., Ergüt, S., Buccella, C., Cecati, C., and Hancke, G. P. (2011). Smart grid technologies: communication technologies and standards. *IEEE transactions on Industrial informatics*, 7(4):529–539.

[Ha, H., Wang, Z. and Lee, J, 2014] Ha, H., Wang, Z. and Lee, J (Springer International Publishing. 2014). Fieldbus network system using dynamic precedence queue (dpq) algorithm in can network. *Intelligent Robotics and Applications*, pages 377–385.

[Hansson et al., 2002] Hansson, H. A., Nolte, T., Norström, C., and Punnekkat, S. (2002). Integrating reliability and timing analysis of can-based systems. *Industrial Electronics, IEEE Transactions on*, 49(6):1240–1250.

[Hanzlik and Kristen, 2013] Hanzlik, A. and Kristen, E. (2013). A methodology for design, validation and performance analysis of vehicle electronic control systems. In Fischer-Wolfarth, J. and Meyer, G., editors, *Advanced Microsystems for Automotive Applications 2013*, Lecture Notes in Mobility, pages 373–384. Springer International Publishing.

[Hartwich, 2012] Hartwich, F. (2012). Can with flexible data-rate. In *13th International CAN Conference (iCC2012), Hambach, Germany*. Citeseer.

[IEC-61375:2-1, 2015] IEC-61375:2-1 (2015). Electronic railway equipment - Train Communication Network (TCN) - part 2-1: Wire Train Bus (WTB).

[IEC-61375:3-1, 2015] IEC-61375:3-1 (2015). Electronic railway equipment - Train Communication Network (TCN) - part 3-1: Multifunction Vehicle Bus (MVB).

[IEC61158:3-6, 2000] IEC61158:3-6 (2000). 61158 parts 3 to 6: Digital data communications for measurement and control: Fieldbus for use in industrial control systems.

[IEC62551, 2012] IEC62551 (2012). Analysis techniques for dependability, petri net techniques. *Geneva*.

[IEEE802.111, 2012] IEEE802.111 (2012). IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, page 35.

[IEEE802.3, 2016] IEEE802.3 (2016). IEEE Standard for Ethernet - Section 1. *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)*, pages 1–628.

[Jensen, 1983] Jensen, K. (1983). *High-level Petri nets*. Springer.

[Kaneshiro et al., 2007] Kaneshiro, P. J. I., Melo, J. I. G., Miyagi, P. E., and Cugnasca, C. E. (2007). Modeling of collision resolution algorithm in lonworks networks. *International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers.*

[Kiefer, 1996] Kiefer, J. (1996). *Methodische Partitionierung und Parametrierung von Feldbussen.* VDI-Verlag.

[Kopetz, 2011] Kopetz, H. (2011). *Real-time systems: design principles for distributed embedded applications.* Springer Science & Business Media.

[Kurczveil et al., 2012] Kurczveil, T., Diekhake, P., Liu, J., and Schnieder, E. (2012). Energy measurement for consumer loads in buildings using field bus system for home automation. In *XX IMEKO World Congress.*

[Kurczveil et al., 2014] Kurczveil, T., Diekhake, P., Liu, J., and Schnieder, E. (2014). Consumer load measurement in automated buildings. *Measurement*, 51:441 – 450.

[LIN, 2003] LIN, C. (2003). Lin specification package, revision 2.0. *Munich, Germany.*

[Lindemann, 1992] Lindemann, C. (1992). *Performance analysis of complex systems by deterministic and stochastic Petri Net models.* na.

[Lindemann, 1998] Lindemann, C. (1998). Performance modelling with deterministic and stochastic petri nets. *ACM SIGMETRICS Performance Evaluation Review*, 26(2):3.

[Liu et al., 2013] Liu, J., Schnieder, E., and Miao, Z. (2013). Message collision and csma/ca modeling and simulation by means of petri net. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on*, pages 413–420.

[Mahmoud, 2000] Mahmoud, R. (2000). Sicherheits-und verfügbarkeitsanalyse komplexer kfz-systeme. *Diss. Universität Siegen.*

[Molloy, 1982] Molloy, M. K. (1982). Performance analysis using stochastic petri nets. *Computers, IEEE Transactions on*, 100(9):913–917.

[Morris, 1986] Morris, H. D. (1986). *Probability and statistics.* Addison-Wesley, Reading Massachusetts, 2nd edition edition.

[Mueller, J. R. and Schnieder, E., 2008] Mueller, J. R. and Schnieder, E. (Valencia, Spain, Sep. 2008). Risk and safety as system-theoretic concepts - a formal view on system-theory by means of petri-nets. *ESREL 2008 - European Safety & Reliability Conference*, pages 2665–2672.

[Munoz-Castaner et al., 2011] Munoz-Castaner, J., Asorey-Cacheda, R., Gil-Castineira, F. J., Gonzalez-Castano, F. J., and Rodriguez-Hernandez, P. S. (2011). A review of aeronautical electronics and its parallelism with automotive electronics. *IEEE Transactions on Industrial Electronics*, 58(7):3090–3100.

[Quiroga, L. M., Becker, U., and Schnieder, E., 2014] Quiroga, L. M., Becker, U., and Schnieder, E. (2014). Das petrinetz modellierungs- und -analysetool π-tool. *at - Automatisierungstechnik*, 62(6):436–445.

[Ramamoorthy and Ho, 1980] Ramamoorthy, C. and Ho, G. S. (1980). Performance evaluation of asynchronous concurrent systems using petri nets. *Software Engineering, IEEE Transactions on*, (5):440–449.

[Reisig, 2012] Reisig, W. (2012). *Petri nets: an introduction*, volume 4. Springer Science & Business Media.

[Ross, 1996] Ross, S. M. (1996). *Stochastic processes.* John Wiley and sons, New York, 2nd edition edition.

[Sauter et al., 2000] Sauter, T., Wollschlaeger, M., and Guericke, O.-v. (2000). Feldbussysteme-historie, eigenschaften und entwicklungstrends. *it-Information Technology*, 42(4):7–16.

[Schnieder, 2013] Schnieder, E. (2013). *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme.* Springer-Verlag.

[Schnieder and Schnieder, 2013] Schnieder, E. and Schnieder, L. (2013). Techniken und methoden. In *Verkehrssicherheit*, pages 317–387. Springer.

[Schnieder et al., 2009] Schnieder, E., Schnieder, L., and Mueller, J. (2009). Conceptual foundation of dependable systems modelling. In *Dependable Control of Discrete Systems*, volume 2, pages 198–202.

[Schnieder, 2008] Schnieder, L. (2008). Towards terminological rigour in the specification of complex automation systems. In *Proceedings of Symposium FORMS/FORMAT*, pages 141–148. Budapest/Hungary.

[Schnieder, 2010] Schnieder, L. (2010). *Formalisierte Terminologien technischer Systeme und ihrer Zuverlässigkeit.* PhD thesis, DLR-Institut für Verkehrssystemtechnik.

[Schnieder, E. and Schnieder, L., 2010] Schnieder, E. and Schnieder, L. (2010). Terminologische präzisierung des systembegriffs. *atp edition-Automatisierungstechnische Praxis*, 52.09:46–59.

[Schrom, 2003] Schrom, H. (2003). *Realisierung eines optimierten Feldbussystems und Modellierung mit Petrinetzen.* PhD thesis, Technische Universität Braunschweig, Braunschweig and Germany.

[Schrom et al., 2011] Schrom, H., Michaels, T., Stein, S., and Ernst, R. (2011). Smallcan-a reliable, low-power and low-cost distributed embedded system for energy efficient building automation. *Energy*, pages 13–18.

[Simon, 2007] Simon, M. K. (2007). *Probability distributions involving Gaussian random variables: A handbook for engineers and scientists.* Springer Science & Business Media.

[Thomesse, 2005] Thomesse, J.-P. (2005). Fieldbus technology in industrial automation. *Proceedings of the IEEE*, 93(6):1073–1101.

[Yiwen Chu, 2013] Yiwen Chu (2013). *Methodische Analyse der unteren Schichten des ISO/OSI Referenzmodells am Beispiel des Feldbussystems SmallCAN.* Master thesis, Technische Universität Braunschweig, Braunschweig, Germany.

[Yurdakul, 2016] Yurdakul, A. (2016). *Morphologisch-semantische Modellierung internationaler Fachsprache. Die sicherheitsrelevante Ortungsterminologie der Landverkehrsfachsprache in Deutsch, Englisch und Türkisch.* PhD thesis, Technische Universität Braunschweig, Braunschweig and Germany.

[Zhu and Schnieder, 2000] Zhu, P. and Schnieder, E. (2000). Holistic modeling of complex systems with petri nets. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 4, pages 3075–3080. IEEE.

[Zhu and Schnieder, 2001] Zhu, P. and Schnieder, E. (2001). Determining traffic delays through simulation. In *Computer-Aided Scheduling of Public Transport*, pages 387–397. Springer.

[Zimmermann and Hildebrandt, 2015] Zimmermann, I. A. and Hildebrandt, A. (2015). Zuverlässigkeitsbewertung komplexer systeme mit stochastischen petrinetzen und timenet. *Technische Zuverlässigkeit, TZZ-VDI.*

[zu Hörste et al., 2013] zu Hörste, M. M., Hungar, H., and Schnieder, E. (2013). Modelling functionality of train control systems using petri nets. *Towards a Formal Methods Body of Knowledge for Railway Control and Safety Systems*, page 46.

List of Figures

List of Tables

Index of Abbreviations

General Abbreviations

Name	Type
APL	Aapplication Layer
BACnet	Building Automation and Control Network
BMW	Beschreibungsmittel, Methoden, Werkzeuge
CAN	Controller Area Network
CPN	Colored Petri net
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DLL	Data Link Layer
DSPN	Deterministic and Stochastic Petri Net
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESD	Electrical Short Distance
FSF	Free-located Special Function
HVAC	Heating, Ventilation, and Air Conditioning
ICE	Intercity-Express
IDE	Identifier Extension Bit
IEC	International Electrotechnical Commission
IFS	Intermission Frame Space
IP	Internet Protocol
KNX	Konnex, a fieldbus protocol for building automation
LIN	Local Interconnect Network
LLC	Logical Link Control
MAC	Medium Access Control
MVB	Multifunction Vehicle Bus
OGF	Optical Glass Fibers
OSI model	Open System Interconnection model
PDF	Probability Density Function
PLC	Programmable Logic Controller
Profibus	Process Field Bus
PROFINET	Process Field Net
QoS	Quality of Service
RAMS	Reliability, Availability, Maintainability and Safety

RTR	Remote Transmission Request
SF	Special Function
SPN	Stochastic Petri Net
TCN	Train Communication Network
TCP	Transmission Control Protocol
UML	Unified Modeling Language
UMLCD	Unified Modeling Language Class Diagram
WTB	Wire Train Bus

Project Abbreviations

Name	Explanation
SmallCAN (Support code: WA3-80029179)	Integrated building automation through a unified open source fieldbus system with low-power and low-cost, funded by European Regional Development Fund.
DIGAFLEX (Support code: 03ET1016A)	A integrated system demonstration in building automation with low-power, low-cost, flexible range of field devices and configuration, funded by Federal Ministry for Economic Affairs and Energy and Energy.
EnEff:Stadt (Support code: 03ET1159A)	Develop a methodology for integrated simulation, management and monitoring of buildings, equipments and local infrastructure networks (gas, electricity and heat), funded by Federal Ministry for Economic Affairs.